

WIPFG

working papers in functional grammar

WPPG no. 24
October 1987

Constructing an FG lexicon on the basis of LDOCE
Hetty Voogt
University of Amsterdam

Constructing an FG lexicon
on the basis of LDOCE

Hetty Voogt-van Zutphen

CONTENTS

	page
Chapter 1. Introduction	
1.1. Aim.....	1
1.2. An FG lexicon.....	2
1.2.1. Predicates.....	4
1.2.2. Terms.....	4
1.3. LDOCE.....	4
1.4. Method and its consequences.....	5
1.5. Structure of the paper.....	6
Chapter 2. Preliminaries	
2.1. Global description of the EXTRACT_LONGMAN program	8
2.2. First steps.....	9
2.3. Contents of LDOCE.....	9
2.4. An FG lexicon on the basis of LDOCE.....	10
2.4.1. Predicates and terms.....	11
2.4.2. Category.....	11
2.4.3. Selection restrictions.....	13
2.4.4. Number and type of arguments.....	13
2.5. The EXTRACT_LONGMAN_FG program	14
2.6. Three remarks.....	15
Chapter 3. The grammatical codes	
3.1. The verb codes.....	18
3.1.1. Introduction.....	18
3.1.2. Rewriting the verb codes.....	19
3.1.3. Discussion.....	21
3.1.3.1. The 0- and 1-codes.....	21
3.1.3.2. The 2-code.....	22
3.1.3.3. The 3-code.....	23
3.1.3.4. The 4-code.....	27
3.1.3.5. The 5-code.....	31
3.1.3.6. The 6-code.....	31
3.1.3.7. The 7-code.....	32
3.1.3.8. The 8-code.....	32
3.1.3.9. The 9-code.....	33
3.1.4. Some remarks on the V-class.....	33
3.1.5. The X-code.....	34
3.1.6. The L-codes: linking verbs.....	36
3.2. The noun codes.....	38
3.2.1. Nominal Predicates.....	38
3.2.2. Terms.....	41
3.3. The adjective codes.....	43
3.4. The grammatical codes: conclusion.....	45
3.4.1. Types of entries in the FG lexicon.....	45
3.4.2. Frequencies.....	47
3.4.3. The relation between LDOCE codes and FG structures.....	48

Chapter 4. The complement words	
4.1. Introduction.....	53
4.2. Semantic functions and prepositions in FG.....	53
4.3. Complement words and intransitive and transitive verbs.....	55
4.3.1. One argument too few.....	56
4.3.2. One argument too many.....	57
4.4. Interim conclusions.....	57
4.5. Verbs with D-codes: complement words do indicate semantic functions.....	58
4.5.1. <i>To</i>	58
4.5.2. <i>For</i>	60
4.5.3. <i>With</i> plus other complement words.....	60
Chapter 5. Selection restrictions	
5.1. Introduction.....	64
5.2. Theory.....	65
5.3. Selection restrictions in Functional Grammar.....	65
5.4. Aarts.....	66
5.5. HPRIM features and semantic codes.....	67
5.6. LPRIM features and subject codes.....	74
5.7. Conclusion.....	75
Chapter 6. Conclusion	
6.1. Constructing an FG lexicon: summary.....	78
6.2. Constructing an FG lexicon: The EXTRACT_LONGMAN_FG program	79
6.3. The future.....	79

Chapter 1. Introduction

1.1. Aim

In this paper, I will attempt to create sets of basic predicate frames and basic terms for a lexicon of English which is intended to fit into the system of Functional Grammar (henceforth: FG) as developed by Dik (1978, 1980, 1983 etc.). There are two ways in which this aim could conceivably be achieved. The entries might be created by hand, possibly with reference to an existing dictionary, but without making any extensive or systematic use of it. On the other hand, one might prefer to rely much more heavily on an existing dictionary, and create entries on the basis of information presented there.

The first method would probably be more time-consuming than the second. It has to be decided which entries are to be included, and all the syntactic and semantic properties of each entry have to be determined. The result of this method is a lexicon which contains entries which are correct with respect to the present state of the theory of FG. Considering the available time, however, it would be fairly small.

With the second method one could use all entries present in a dictionary. Moreover, a good dictionary contains information on syntactic and, possibly, on semantic properties of its entries. In the ideal case, this information matches exactly the information needed in the predicate frames and term structures in the FG lexicon. If there is no complete match, however, the differences between the information found in the dictionary and that needed in Functional Grammar have to be stated, and, in creating the FG lexicon, information may have to be added, deleted, or changed. Whether the dictionary information matches the information used in the FG-lexicon or not, rules have to be postulated which rewrite the available information into the format used in Functional Grammar.

If these rewrite rules are used by a computer program which creates FG-entries on the basis of information from an existing dictionary, one can rewrite all entries in the dictionary just as easily as only a few examples. If such a program runs over an entire dictionary, the result is a very large lexicon. The value of this lexicon depends on the correctness of the rules. A rewrite rule is correct if it always leads to correct predicate frames or term structures. Since it would be impossible, within the scope of this paper, to check every entry in such a full-scale FG lexicon, one could never say with any certainty that all entries in the lexicon are correct with respect to the theory of FG. The result of the second method, therefore, would be a large lexicon, the entries in which are not necessarily all correct.

Up till now, lexicons used to test or demonstrate (parts of) FG were written according to the first method in a rather *ad hoc* fashion. They always contained a fairly

small number of entries. In this paper, I will attempt to construct a full-scale FG lexicon by the second method. The dictionary which will be used is the *Longman Dictionary of Contemporary English*¹ (henceforth LDOCE), which is available on computer tape.

1.2. An FG lexicon

The lexicon of Functional Grammar consists of:

- a) a set of basic predicate frames and a set of basic term structures;
- b) one or more meaning postulates for each predicate frame and each term.

(Cf. Dik, 1986: 15-27)

As we saw above, my aim is to create the sets of basic predicate-frames and term structures. I will not concern myself with the **meaning postulates**, since treating them properly requires a study in its own right. The main reason for this is that constructing these postulates in accordance with the theory set forth in Dik (1978b) would require one to interpret and entirely rewrite all information on meaning from an existing dictionary. A project is currently under way in which an attempt is made to create hierarchical meaning chains which are very similar to those proposed in *Stepwise Lexical Decomposition* on the basis of LDOCE.² Thus, I will only be concerned with the creation of sets of predicate-frames and term structures.

The role that the lexicon plays in the total system can be seen in the Figure 1 (from S.C. Dik, 1978a). Since the lexicon contains only **basic** predicates and terms, as can be seen in this Figure, I will not be concerned with derived predicates and terms. All main entries in LDOCE will be regarded as basic (See §2.3). All derived forms (by 'derived' I mean: formed by a synchronically productive process) should be created by predicate-formation, and should not be found in the lexicon. For example, the form *paint with* will not be in the lexicon, since it can be derived from the form *paint on* by a rule of Complete Verb Formation (see § 4.5.3.).

Whenever some derived form is internalized by a language user, when a form may be seen as the result of a rule which is no longer productive, or when a derived form starts to lead its own life by taking on a more specialized or idiomatic meaning, it is added to the total store of predicate frames or term structures in the lexicon. This means that the language user does not have to go through the process of derivation each time the form is encountered, but can pick the form directly from the lexicon. The lexicon, therefore, contains all predicates and terms of the language which are not derived from other predicates or terms by synchronically productive rules (Cf. Dik, 1980: Ch. 2, 1986)).

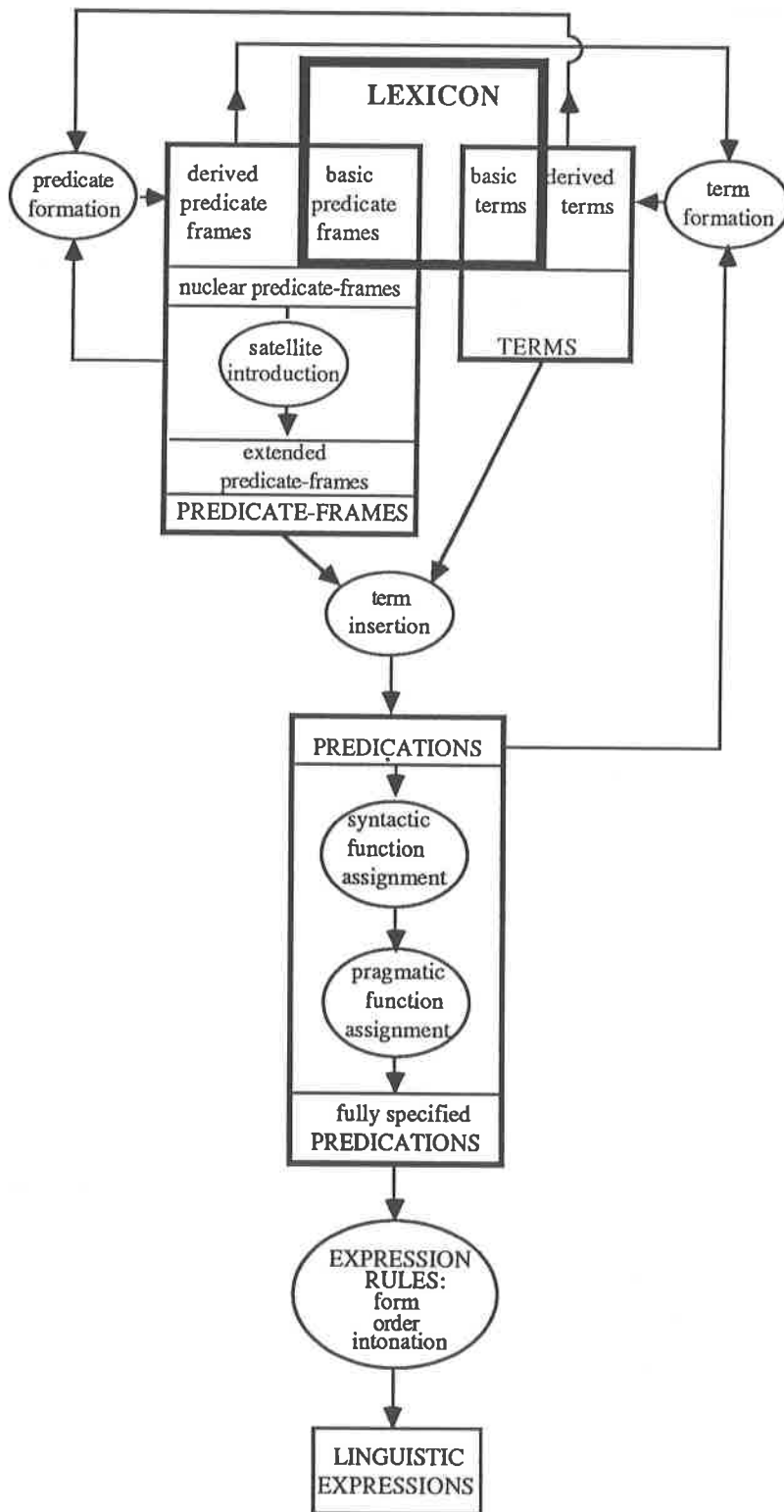


Figure 1. An overview of FG

1.2.1. Predicates

Predicates are contained in predicate frames. Predicates are 'expressions which designate properties of, or relations between entities' (Dik, 1983: § 5.1.). Predicate frames specify: (i) the predicate; (ii) the syntactic category of the predicate; (iii) the argument positions associated with the predicate indicated by variables x_1, x_2 , etc.; (iv) the semantic functions associated with these argument positions; and (v) the selection restrictions imposed on these argument positions (ibid.):

$$\text{Predicate}_{\text{Category}}(x_1: \langle \text{restriction} \rangle(x_1))_{\text{function}} \dots (x_n: \langle \text{restriction} \rangle(x_n))_{\text{function}}$$

1.2.2. Terms

Terms are contained in term structures. They are 'expressions which can be used to refer to entities in a person's mental Picture' (Dik, personal communication). Basic terms consist of: (i) the term; (ii) the syntactic category of the term; (iii) the argument position associated with the term, indicated by variable x_1 ; a term operator or a combination of term operators ω :

$$(\omega x_1: \text{predicate}_{\text{Category}}(x_1))$$

Predications are formed when terms are inserted into the argument positions of a predicate frame. Predications are taken to designate sets of states of affairs.

1.3. LDOCE

The lexicon will be constructed on the basis of the machine-readable version of the *Longman Dictionary of Contemporary English*. There are several reasons why I chose Longman's dictionary as the basis for this project. Two of the reasons are substantial in that they relate to the form and contents of the dictionary; two other reasons are more or less coincidental in that they are both related to my work on the ASCOT project.³

I will start with the substantial reasons. In a comparison between the LDOCE-tape and the machine-readable version of the *Oxford Advanced Learner's Dictionary* (OALD), which are the two dictionaries that we have on computer-tape, it was found that the LDOCE-tape was to be preferred to OALD (Akkerman et al., 1985). However, the ASCOT-team could not simply work with LDOCE without encountering any problems. One of the reasons for some of our problems may have been that the tape was not originally meant as input to computer programs, but was used as a type-setting tape and a medium for storage which could easily be changed or updated.

Some of LDOCE's features which had to be taken care of were the following. The tape contains numerous setting instructions (even though fewer than in OALD). It

contains errors, such as a missing wordclass. And, finally, it contains inconsistencies: thus, to a human reader the wordclass information *comb form* and *comb. form* may seem the same, but for the computer these two are simply distinct because one has a period and the other does not. Unless one takes care of all sorts of special features/irregularities/mistakes beforehand, one's programs will run into errors time and again.

The second substantial reason for choosing LDOCE is the fact that this dictionary on computer tape makes use of a system of semantic coding. Some of these codes are comparable to selection restrictions. One of the codes, for instance, indicates that a verb takes a human subject. These codes are not found in the dictionary in its printed form, probably because they have not yet been sufficiently worked out. I think that these semantic codes, which no dictionary I know of includes, may be very useful.

The last two reasons why I chose Longman's dictionary are: 1) in the ASCOT project we have, among other things, studied the structure of LDOCE, and critically assessed its grammatical coding system, and 2) a huge program has been written to extract the information from LDOCE. Some parts of this program will not be necessary for the present project, and some procedures will have to be added, but on the whole, a lot of work has been done which is very useful for present purposes.

1.4. Method and its consequences

Creating the sets of predicate frames and term structures in an FG lexicon can be called neither a small, nor a well-delimited task. I do not even imagine that it can be handled entirely in this paper. However, I would like to see how far I can get in creating these structures with very **general** rules extracting **some** of the information contained in LDOCE. It will be appreciated that the rules that may be used for extracting information from a reasonably conventional dictionary like LDOCE and for rewriting this into FG-like structures are not necessarily self-evident. I will try and describe as explicitly as possible what information from the dictionary I use, and what I do to it in order to turn it into something that satisfies the conditions set by Functional Grammar.

Often, this method means that I cannot go into matters in very much detail. It will frequently be found that much more thorough investigation is necessary to reach a satisfactory understanding of a problem. With the approach that I have opted for, I am forced to be content with research which goes into very little detail. However, some advantages that my approach has are:

- a large amount of data will become available for further research (see §1.1);
- since I will make them explicit, people may challenge my rules and change or refine their results.

In creating my lexicon, I try as much as possible to take into consideration structures that have been proposed in earlier publications on Functional Grammar.

1.5. Structure of the paper

The next chapters in this paper will investigate how the information found in LDOCE can be used to create entries in an FG lexicon. In chapter 2, I will study exactly what LDOCE has to offer and what types of information are needed in the FG lexicon. I will compare the types of information needed by the FG lexicon and those offered by LDOCE and study how the extraction of some types of information may be brought about. In chapter 3, I will work out how the so-called grammatical codes in LDOCE can be used to predict the number and type of the arguments of verbal, nominal and adjectival predicates, and the structure of terms. Chapter 4 shows that, sometimes, complement words given in LDOCE may be used to further refine the predicate frames developed in the preceding chapter, in particular by enabling one to add semantic functions to the arguments in the predicate frames. However, it will also become clear that, sometimes, the coding system used in LDOCE leads to structures which are unacceptable from an FG-perspective.

In chapter 5, I will be concerned with the selection restrictions found on the arguments in the FG lexicon. LDOCE gives for each sense of an entry a number of 'semantic codes'. Some of these are exactly the selection restrictions needed in Functional Grammar. I will show which semantic codes LDOCE uses for this purpose. I will also try to show that working with selection restrictions will not always be an easy matter. In this context, I will refer to Aarts (1976), who has worked a system to describe adjective-noun combinations making use of features. Among these, we find contextual features, called HPRIM and LPRIM features, which are comparable to selection restrictions. I will indicate that the subject codes (indicating semantic fields) given in LDOCE might be used to supplement the semantic codes.

In the last chapter, chapter 6, I will draw conclusions from my investigations and summarize my findings. The main conclusion will be that a lot of work still needs to be done to arrive at acceptable predicate frames and term-structures in an FG lexicon.

Notes to Chapter 1

- ¹ There is copyright on all material from the *Longman Dictionary of Contemporary English*. This means that no part of the information from the dictionary may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior permission of the Copyright owner (except for publication of the results of research relating to the use of the dictionary in academic publications). The files created on the basis of the dictionary are subject to the same restrictions.
- ² This project, called 'LINKS in the Lexicon', is funded by the Organisation for Pure Academic Research (ZWO). The project leader is Dr. W.J. Meijs. For a description of the aims of the LINKS-project, see Meijs, (1985).
- ³ The ASCOT-project is a project funded by the Dutch organisation for Pure Academic Research (ZWO) from March 1st 1984 until March 1st 1987. The last four months of the project (March 1st until July 1st 1987) were financed by the Faculty of Arts of the University of Amsterdam. The aims of the project were to build a lexical database on the basis of LDOCE and a system to code words in a text grammatically. For a full description of the aims of the project see Meijs, (1984). Reports on the project can be found in Akkerman et al. (1985) and Akkerman et al. (forthcoming)

Chapter 2. Preliminaries

2.1. Global description of the EXTRACT_LONGMAN program

EXTRACT_LONGMAN is the name of the computer program that was written (in PASCAL) to extract the information needed for the ASCOT-project from the computer-tape version of LDOCE. Basically, it was written to extract and interpret the information on the tape and put it in an optimally-structured format. This program is essential for correct interpretation of each piece of information provided by LDOCE and, subsequently, for further use of the contents of the dictionary.

I will here summarize the most important procedures of EXTRACT_LONGMAN:

- creation of new entries for 'also-words' and American English variants, copying the information given with the original entry, e.g.

- (1) a. *creativity* also *creativeness* /.../ n [U]
b. *colour*, AmE *color* v [T1; X(1),7] (...) 2 [I0] (...) etc.

- creation of full forms of derivations of which only the affix is given, e.g.

- (2) a. tilde: *drunken* (...) ~*ly* adv ~*ness* n [U]
b. hyphen: *evangelist* (...) -*lism* n [U] -*listic* adj

- creation of separate entries for words with more than one wordclass in the same entry (See § 2.4.2. below), e.g.

- (3) a. *satin* n, adj [Wa5; U; B]
b. *depilatory* adj, n [Wa5; C; U]

- creation of one character for each wordclass (See §2.4.2 below)

- treatment of spelling information and irregular forms, e.g.

- (4) a. *picnic* v -*ck*- (...)
b. *colossus* n -*suses* or -*si*
c. *drink* /.../ v *drank* /.../, *drunk* /.../ (...)

- decomposition of complex codes, e.g.

- (5) a. *desire* v [T1,3,5c;V3]
b. *break off* v adv [T1,4;I0: (with)]
c. *see* v [Wv6;T1,5a,6a;V2,3 (fm) & only pass.), 4,8]

- treatment of syllable divisions, e.g.

- (6) *grad.u.a.tion* /.../ n (...)

- addition of default codes for nouns and adjectives (C and B)

- distinction of code types, e.g.

- (7) *walrus* n [Wn1] 1 [C] (...) 2 [A] (...)
[Wn1] is a special feature code, [C] is a countability code, [A] is a position code (See Appendix 2)

2.2. First Steps

The first task in building a FG lexicon on the basis on LDOCE is to extract all types of information (such as lemma and wordclass) from the dictionary and decide whether the information is **useful** for our purposes or not.

If it is decided that the information will be used, then one out of three possible situations will occur. The information may be used directly (**direct method**) or more information may be needed in order for the present bit of information to be useful. In such cases, extra information may be either found elsewhere among the information given in LDOCE (**indirect method**) or it may have to be added manually (**manual method**). Of course, it may also occur that information is needed for the construction of an FG-entry which is not found in LDOCE at all. This information will have to be added by hand (**manual method**).

As the aim of this paper is to extract information from LDOCE by rule, I am not interested in the manual method; nor am I interested in cases which require examination of large numbers of examples in order to arrive at a justified decision. This is a consequence of the fact that with the approach I have opted for, I have to be content with research that goes into little detail (Cf §1.4.).

2.3. Contents of LDOCE

The machine-readable version of Longman's dictionary is structured as follows. The information is contained in lines. Each line has a number code which specifies the position on the tape and the type(s) of information it contains. The numbers indicating type of information are used as follows:

Number	Type(s) of information (each type starts with '-')
1:	- lemma with syllabic divisions (if there is more than one entry with the same spelling (i.e. if there are any homographs), syllabic divisions are only given for the first homograph) - variant spelling forms
2:	- homograph number - entry type - entry form
3:	- pronunciation and stress (only for the first homograph)
4:	- also-word (= alternative form with the same meaning. E.g. <i>yeti</i> - also <i>abominable snowman</i>)
5:	- part of speech (\approx wordclass) - spelling information (e.g. irregular forms)
6:	- grammatical codes for the entry in all its senses - label (\approx register and type of English (e.g. British English))
7:	- sense number - grammatical codes which hold for the sense indicated by sense number - subject code of the present sense (semantic field) - semantic codes of the present sense

- 8: - meaning, or an expression that the lemma is part of plus its meaning;
 - examples of a form of the lemma used in context
 18: - continuation of 8
 9: - usage
 19: - continuation of 9
 10: - subheadword
 - pronunciation of subheadword
 - grammatical codes of subheadword
 11: - illustration size
 20: - beginning of a section containing words starting with the same letter

Whenever a number is used for more than one type of information, then all these types of information are separated from each other by the delimiter '<'. An example of a line which has number 7 is:

(8) 12225207<0200<Wv5;I0<GOZL< X

The last number before the first '<' (7) indicates what types of information the line contains, the homograph number is 2, the grammatical codes are Wv5 and I0, the subject code is GOZL, and the semantic codes are: ' X ' (9 spaces and an X).

Although all types of information are interesting to have in an FG lexicon, and have been identified and interpreted by the EXTRACT_LONGMAN program, I want to limit myself to using only those types of information that are absolutely necessary for the construction of entries in the FG lexicon. I will not deal with the entry type and entry form, (which give information about the internal structure of compound forms, such as the part of speech of the first and the last part and the stress pattern of the parts of the compound), homograph number, pronunciation and stress, syllable division, usage, idiomatic expressions, and most important of all, I will not deal with the meaning of entries in so far as their definition is concerned. Variant spelling forms and also-words will be handled as follows: they will get the same information (wordclass, grammatical and semantic codes) as the lemma under which they are found. Subheadwords will not be included in my lexicon for two reasons. In the first place, no semantic codes are given for subheadwords, so the entries to be created would be incomplete anyway. Secondly, most subheadwords are forms which have been derived from the headword, and derivations should not be included in the lexicon, but should be created in a morphological component.¹ An example of a subheadword that has been left out is the form *sweetness*, which is found in the entry for *sweet*. Basically, what I will use for each entry are the wordclass, the grammatical codes, and the semantic codes.

2.4. An FG lexicon on the basis of LDOCE

The way in which each type of information in the predicate frame (cf. §1.2.1.) is found in LDOCE can be seen in the following figure:

FG-info	found in LDOCE as	linekind in LDOCE
Predicate	'lemma'	1
Category	'part of speech'	5
Sel(ection) Restr(iction)	'semantic code'	7
$x_1 \dots x_n$	'grammatical code'	6, 7
Fu(nction)	is not found as such in LDOCE.	

Whereas the predicate, the category, the selection restrictions, and the number and type of the arguments ($x_1 \dots x_n$) can be drawn from the dictionary by the direct method, this is not the case for semantic functions. As we will see later, semantic functions can only very rarely be deduced by combining pieces of information in LDOCE (indirect method). I am afraid that they can often only be added by hand.

An example of a basic term is:

$$(9) (d1x_1: he_p(x_1))$$

Basic terms are pronouns and proper names. I will use from LDOCE entries with wordclass pronoun and grammatical code Wp1 or Wp2, and nouns with subtype R ('namelike noun') (indirect method). Proper names, which are also basic terms in Functional Grammar, are not found in LDOCE.

In the following sections, I will look in more detail at the way in which each type of information in the FG lexicon will be extracted from LDOCE.

2.4.1. *Predicates and terms*

Nothing will be changed in the information found in LDOCE. Spaces in the entries are allowed in the case of multi-words.

2.4.2. *Category*

The following parts of speech are found in LDOCE: n, adj, v, adv, pron, prep, det, predet, suffix, prefix, comb. form, v prep, v adv, v adv prep, conj, interj, and other (abbrev. for, contr. of).

The FG lexicon has terms and nominal, adjectival and verbal predicates (Dik, 1978: 29). From LDOCE nouns (n) with code R and pronouns (pron) with grammatical code Wp1 or Wp2² will be used as basic terms.³ Nouns (n) other than those with code R, adjectives (adj), verbs (v), prepositional verbs (v prep), phrasal verbs (v adv) and phrasal prepositional verbs (v adv prep) will be used as basic predicates. Since it would be a pity to 'throw away' all other information, I will put this in a separate output file. One particularly likely candidate for insertion in the FG lexicon at a later stage would be

adverbs. Many elements which can function as adverbs in a sentence can be formed by term formation, such as *the next day* in:

(10) They left on the next day.

Elements like *yesterday* and *tomorrow*, however, cannot be formed by term formation and will always function as adverbs in a sentence. They should therefore be included in the lexicon.

The EXTRACT-LONGMAN program has done two things to the wordclass information in the dictionary:

- it has changed all wordclasses into one character as follows:

Used in FG lexicon		Not used in FG lexicon	
LDOCE class	ASCOT class	LDOCE class	ASCOT class
n	N	adverb	R
adj	A	conjunction	K
v	V	determiner	D
v prep	J	predeterminer	Z
v adv	F	prefix	E
v adv prep	G	prep	O
pron	P	comb. form	L
		suffix	U
		interjection	I
		other:	X

- from one entry with two wordclasses it has made two entries with one wordclass each. E.g. *ablative adj, n* has become *ablative A* and *ablative N*

Prepositional verbs and phrasal prepositional verbs need special attention. It was suggested by Simon Dik (personal communication) that prepositional verbs should be treated as follows: the preposition should be separated from the verb (plus adverb) and inserted into the predicate frame with one of the arguments. However, arguments against this view are:

- 1) The argument and the preposition may be separated from each other.

(11) The matter was looked into

- 2) The prepositional verb can often be replaced by a verb without a preposition, because, semantically, it forms a unit

(12) a. looked into the matter.

b. They considered the matter

I will therefore not take specific steps in handling these verbs. I will keep the component parts of the prepositional and phrasal prepositional verbs together and include the verbs (+ adverb) + preposition as predicates with the appropriate wordclass in my lexicon. In this way they will be easy to find for anyone who wants to study this type of verbal structure. Thus, *look into* would be represented as follows

(13) look into_J (x₁) (x₂)

2.4.3. Selection Restrictions

Selection restrictions will be incorporated in the FG-entry as they are found in LDOCE. I have included the list of semantic codes used by Longman in Appendix 1. I will go into the selection restrictions in chapter 5.

2.4.4. Number and Type of arguments

The number and type of the arguments will be deduced from the grammatical codes found in LDOCE. For a list of all grammatical codes, see Appendix 2. Each grammatical code contains up to three components. The first of these components, which must be present, is a big letter code. Especially for nouns and adjectives, only a big letter is often found. With verbs in particular, however, the big letter is often followed by a number. The numbers give information on complementation. For instance, a number may tell us that a verb takes a subclause and what form the subclause has, it may tell us the category of the complement etc. Finally, the number may be followed by a third component: a little letter. The little letters give information which is more specific still than that of the numbers, and make sense only in combination with a number. One of the little letter codes indicates, for instance, that the subordinator *that* may be left out before a subclause.

The W-codes are particularly difficult to deal with. These W-codes (except Wp1 and Wp2) give information that I cannot easily think of ways of dealing with or they give information which I do not think is absolutely necessary for the construction of my predicate frames. The Wv6-code, for instance, which says that a verb is not used in the *-ing*-form, might be used to indicate that a verb is -DYNAMIC. However, it is not the case that only this code supplies that piece of information. Often one finds that this information is given informally, for instance spelt out as 'no passive', instead of the code Wv6. I find this too difficult to handle adequately here. I will therefore disregard the W-codes for the moment.

As regards the little letter codes, I find the information supplied by [1a, 4a, 1b, 5b] either too informal, or unnecessary for building the predicate frames. However, I will use [4b, 5a, 5c, 6a, 6b].

In chapter 3, I will show how the grammatical codes will be used in the construction of predicate frames and term structures (See also Akkerman et al. (forthcoming)).

2.5. The *EXTRACT_LONGMAN_FG* program

The *EXTRACT_LONGMAN* program was taken as the basis for creating a set of files which would serve as test-files for my FG lexicon. For this purpose, the program was changed in a number of ways, among others as indicated in §2.3 (only wordclasses *n*, *adj*, *v*, *v adv*, *v prep*, *v adv prep* and *pron* are used; subheadwords are not handled, etc.). The version of *EXTRACT_LONGMAN* containing these changes was called *EXTRACT_LONGMAN_FG*. The first files created by this program were in a format that allowed pattern-matching by the program *QUERY*.⁴ In Appendix 3, I will show how the information in the *QUERY*-files is represented. The tests that could be performed on these files were the following:

- 1) the contents of the files could be inspected to see whether the information extracted from the dictionary is the information we are interested in;
- 2) the contents of the dictionary could be checked: the number of occurrences of specific grammatical or semantic codes could be counted, as well as the number of occurrences of combinations of codes, etc. The data thus made available were used as the basis for the rewrite rules (Cf chapters 3, 4, and 5). All statistics in this paper were found using *QUERY* with these files.⁵

The most important change that I made in the *EXTRACT_LONGMAN* program (beside the changes mentioned in §2.3) was that the semantic codes were included and attached to the grammatical code(s) to which they belonged. As can be seen in §2.3., both the semantic codes and the grammatical codes are found in lines marked by the number 7. Lines marked 7 always precede, and apply to the meaning-information in the line(s) marked 8 or 18. Each sense of an entry is thus preceded by information on its grammatical and semantic characteristics. Whereas in the *ASCOT*-project we were only interested in over-all grammatical properties of entries, and not in semantic properties of entries, in an FG lexicon this semantic information is important. Therefore, I extracted from each line marked 7 the grammatical and semantic codes and related the semantic codes to the grammatical codes. The way in which this was done will be shown using the following example, which was found in the entry for the verb *accept*. This entry has two codes (transitive and intransitive) for the same sense:

(14) 00144607<0100<T1; IO< < H Z

In this line marked 7 (number before first '<'), which precedes the first sense of the entry (0100),⁶ the grammatical codes T1 and IO are found; the semantic code for subject is H (human) and the semantic code for the object is Z (unmarked). The entry in the query-files created by the *EXTRACT_LONGMAN_FG* program contains the following elements: 'T1|H Z' and 'IO|H Z'.

2.6. Three remarks

I want to make three remarks before I start off rewriting the grammatical codes for my (FG-)purposes. The first is that codes that LDOCE uses are often comparable to or identical with the subclassification that is used in Quirk et al. (1985), *A Comprehensive Grammar of the English Language* (CGEL). This is not surprising if we consider the fact that both Quirk and Leech, who were co-authors of CGEL, contributed advice and suggestions for LDOCE. I will sometimes show similarity or differences between data from LDOCE and from CGEL.

The second remark I would like to make is that whenever I show an example of an entry that has a certain grammatical code, this does not mean that that grammatical code is the **only** grammatical code that is given for that particular word. Usually, an entry in LDOCE has various grammatical codes. For instance, the entry for the verb *grow* has codes I0 and T1, used in different senses, which means that the verb can both be used transitively (followed by one or more nouns or pronouns) and intransitively. Examples of the two uses are:

- (15) I0: Grass grows after rain
- (16) T1: He grows vegetables

My last remark before starting my investigation concerns the data I use. In the following chapters, I will investigate how grammatical codes in LDOCE can be used to create predicate frames and term structures of entries in an FG lexicon. The problem is how to rewrite the available information so that structures are created that can grow into fully-specified predicate frames and term structures. It is, of course, impossible to look at every entry in the dictionary that has a certain code in order to arrive at a justified FG-structure for that code. I will mainly look at the examples given to illustrate the codes in the Table of Codes on the back cover of LDOCE in printed format. It is on these examples that I will base an FG-frame. In addition to this, I will randomly select some entries from the dictionary that also have that code and see how they fit in (cf Appendix 4). Whenever I find that the original FG-structure does not cover the present entry, I will create a new structure. I will always use the/an example-word that I base a rewrite rule on in the rewrite rule.

As will be seen in the following chapters, this method may lead to ambiguity of LDOCE codes. However, it may also result in the fact that a certain LDOCE code will not be used in creating the lexicon at all; this is indicated by NOT IN LEXICON. I will go into the criteria upon which I base my decisions in the relevant sections. I will also, at times, mention characteristics of certain groups of words.

Notes to Chapter 2

¹ Although many derived forms are included in LDOCE as subheadwords, some forms which can be argued to be derived (e.g. *partnership*, *commitment*) are given as main entries (headwords or 'also'-words). As I include all main entries from LDOCE in my lexicon, the lexicon will unfortunately include some derived forms.

² LDOCE uses the wordclass *pron* for various types of words: determiners (*all*, *some*), numerals (*eight*, *sixty*), 'real' pronouns (*he*: personal, *who*: relative), and other (*anything*). Apparently, the possibility of occurring in isolation is regarded as their defining characteristic:

He asked for money and I gave him some.

I think that we are not dealing with the pronoun *some* here, but with a determiner (that is, with the expression of a term operator) which is used with an unspecified term. I will not include such elements in my lexicon.

However, I do want to include the entries which I have called 'real' pronouns above. Fortunately, the personal and relative pronouns are distinguished from the other pronouns: they have the grammatical codes *Wp1* and *Wp2*, respectively, whereas the other entries with wordclass *pron* do not have any grammatical codes.

³ It is not quite clear to me how the pronouns should be treated in the lexicon. In Van der Korst (1987), it is argued that pronouns are not ready-made, basic terms, because, at the moment of insertion, the pronominal case form in the sentence is not yet determined. On the level of the predication these elements are presented as follows:

- ([+Speaker, -Hearer] 1x₁) will lead to *I/me* in the linguistic expression
- ([+Speaker, -Hearer] mx₁) will lead to *we/us* in the linguistic expression
- ([-Speaker, +Hearer] 1x₁) will lead to *you* (singular)
- ([-Speaker, +Hearer] mx₁) will lead to *you* (plural)
- ([-Speaker, -Hearer] 1x₁) will lead to deictic *he/she/it/him/her*
- ([-Speaker, -Hearer] mx₁) will lead to deictic *they/them*
- (A1x₁) will lead to anaphoric *he/she/it/him/her*
- (Amx₁) will lead to anaphoric *they/them*
- (R1x₁) will lead to relative *who/whom/which* (singular)
- (Rmx₁) will lead to relative *who/whom/which* (plural)

Presumably, the pronouns should be included in the lexicon with these or similar structures. This will have to be changed by hand, however, since no indication is given in LDOCE by which these forms could be distinguished from each other automatically. I will include the personal and relative pronouns in my lexicon as follows: e.g.

(i/dnx₁: he_p(x₁))

⁴ The software for the QUERY-system was designed and written by G.J. van der Steen. See Van der Steen (1984).

⁵ Unfortunately, the output of QUERY is not always entirely reliable. The numbers found in the tables and text of chapters 3, 4, and 5 should, therefore, not be taken as absolute but as approximate numbers.

⁶ This sense is described in the following lines on the tape (marked 8 and 18) as: 'to take or receive (something offered or given), esp. willingly; to receive with favour'.

Chapter 3. The Grammatical Codes

3.1. The Verb Codes

3.1.1. Introduction

The verb codes are used in LDOCE as shown in the table below. The numbers in this table indicate in how many entries in LDOCE the code is used at least once. Thus, there are 237 entries which have code T4. Note that these numbers are indications of the frequency of each **type** of verbal construction. The numbers tell us nothing about the frequency of the **tokens** of these verbal constructions: a verb with a code which has low frequency in LDOCE may be used very often in verbal interaction. For example, one of the codes for *be* is [I8]. This code is used only twice in LDOCE: for *be* and *have*. With *be*, [I8] is used to indicate its use in the passive construction, a construction which is used very frequently in English.

In examining the table below, it might be interesting to look at the Table of Codes from LDOCE at the same time: a copy of the LDOCE table can be found in Appendix 7.

	D	I	L	T	V	X
0		3513		1		
1	499	1	66	6769		87
2		12		4	15	
3	1	81	3	93	229	
4		9	4	237	114	1
5	33	11	2	262		1
6	13	2	1	129		
7			55	1		124
8		2	1		5	
9			570			476

The following codes should not have been used according to the Table of Codes, but they are found in the dictionary:

Code	Word	Example given in LDOCE	Decision
D3:	<i>trust</i>	You can't trust him to do anything right	OK
I1:	<i>wish</i>	No example given	error
T0:	<i>pop</i>	No example given	error
T7:	<i>argue</i>	(Error in structure of the LDOCE code)	error
X4:	<i>save</i>	It'll save me going into town	should be D4
X5:	<i>warrant</i>	(Error in structure of the LDOCE code)	error

Trust will be found among the rules for D-codes; *save* is found among the rules for X-codes, but is rewritten as D4.

3.1.2. Rewriting the Verb Codes

In this section each grammatical code from LDOCE is found together with the predicate frame that an entry having this code will get in my lexicon. I will start the rewrite rules for each 'big letter code' with a description of the meaning of this code taken from the introduction to LDOCE.¹

[D]: "verbs that are followed by 2 nouns, pronouns, or nounlike expressions, which are often used as indirect object and direct object. These nouns, pronouns, etc. always represent something else, not each other".

D1: give_v [D1] (x₁) (x₂) (x₃)

D3: trust_v [D3] (x₁) (x₂) (x₃: [Inf PRED .. (x₂) ..] (x₃))²

D5: advise_v [D5] (x₁) (x₂) (sub x₃: [PRED] (x₃))

D5a: warn_v [D5a] (x₁) (x₂) ((sub) x₃: [PRED] (x₃))

D5c: petition_v [D5c] (x₁) (x₂) (x₃: [SUBJUNCTIVE PRED] (x₃))

D6a: tell_v [D6a] (x₁) (x₂) (wh x₃: [PRED] (x₃))

D6b: tell_v [D6b] (x₁) (x₂) (wh x₃: [Inf PRED.. (x₂) ..] (x₃))

[I]: "verbs, most of which are intransitives, that need not be followed by anything".

I0: pause_v [I0] (x₁)

I2: modals and *do*: NOT IN LEXICON

I3: yearn_v [I3] (x₁: [Inf PRED](x₁)) OR
(x₁) (x₂: [Inf PRED .. (x₁) ..](x₂))

I4: come_v etc.: NOT IN LEXICON

burst out_P etc.: NOT IN LEXICON

I5: chance_v [I5] (sub x₁: [PRED (+ FACT)] (x₁))

N.B. grizzle_v etc. should be T5

I5a: appear_v [I5a] ((sub) x₁: [PRED (+FACT)] (x₁))

I6a: appear_v [I6] (sub x₁: [PRED (-FACT)] (x₁))

I8: NOT IN LEXICON

[L]: "Linking verbs, which are followed by something that represents (refers to) the subject. This "something" may be a nounlike expression, or an adjective serving as a complement, or an adverbial".

The L-codes will not be included in the lexicon, except L9:

L9: live_v [I9] (x₁) (x₂)

[T]: "verbs, many of which are transitives, that are followed by a noun or nounlike expression as a direct object which does not represent the subject, unless reflexive".

T1: kick_v [T1] (x₁) (x₂)

T2: help_v [T2] (x₁) {(x₂)} (x₃: [Inf without *to* PRED ..(x₂) ..] (x₃))

N.B. come into_j etc. should be T1

T3: want_v [T3] (x₁) (x₂: [Inf PRED] ..(x₁) ..) (x₂)

T4: hate_v [T4] (x₁) (x₂: [NOMINALIZED PRED .. (x₁) ..] (x₂))

reckon on_j [T4] (x₁) (x₂: [NOMINALIZED PRED .. (x₁) ..] (x₂))

N.B. T4a (e.g. believe in_j) will be treated like T4

T5: pledge_v [T5] (x₁) (sub x₂: [PRED] (x₂))

T5a: know_v [T5a] (x₁) ((sub) x₂: [PRED] (x₂))

T5c: desire_v [T5c] (x₁) (x₂: [SUBJUNCTIVE PRED] (x₂))

T6a: anticipate_v [T6a] (x₁) (*wh* x₂: [PRED] (x₂))

N.B. T6 (e.g. discover) will be treated like T6a

T6b: fix_v [T6b] (x₁) (*wh* x₂: [Inf PRED .. (x₁) ..] (x₂))

[V]: "verbs that are followed by a 2-part direct object. The first part is a nounlike expression, and the second part is an infinitive with or without *to*, an *-ing* form, or a past participle".

V2: see_v [V2] (x₁) (x₂) (x₃: [Inf without *to* PRED ..(x₂) ..] (x₃))

V3: want_v [V3] (x₁) (x₂: [Inf PRED](x₂)) OR

(x₁) (x₂) (x₃: [Inf PRED .. (x₂) ..](x₃))

V4: see_v [V4] (x₁) (x₂: [NOMINALIZED PRED](x₂)) OR

(x₁) (x₂) (x₃: [NOMINALIZED PRED .. (x₂) ..](x₃))

approve of_j [V4] (x₁) (x₂: [NOMINALIZED PRED] (x₂))

N.B. V4a (e.g. rely on_j) will be treated like V4

V4b: talk into_j [V4b] (x₁) (x₂) (x₃: [NOMINALIZED PRED ..(x₂) ..] (x₃))

V8: want_v NOT IN LEXICON

[X]: "verbs that are followed by a nounlike expression as direct object and by a second expression that represents (refers to) the direct object. The second expression is a complement and may be a nounlike expression, an adjective, or an adverbial".

X1: consider_v [X1] (x₁) (x₂: [(Inf) PRED_n] (x₂))

appoint_v [X1] (x₁) (x₂) (x₃) OR

(x₁) (x₂: [(Inf) PRED_n](x₂))

X4: save_v [D4] (x₁) (x₂) (x₃: [NOMINALIZED PRED ..(x₂) ..] (x₃))

X7: consider_v [X7] (x₁) (x₂: [(Inf) PRED_a] (x₂))

X9: put_v [X9] (x₁) (x₂)(x₃)

As yet, I have not included any semantic functions (but see Chapter 4) in these predicate frames. I do not think it is wise to assign these semantic functions without doing some further research, for instance into the classes of verbs that have a certain grammatical code. I am afraid it will be quite difficult to automatically assign semantic functions. The verbs *remain* and *fall*, to give but one example, are both coded IO, but *remain* has a Positioner as argument, whereas the argument of *fall* is Processed.

3.1.3. Discussion

In this section I will discuss the changes I proposed in the last section. I will go through the changes by number. In this way big letters with the same number will be discussed in the same subsection. The discussions of the linking verbs and the X-verbs will be found in separate sections, as well as some remarks on the V-verbs.

3.1.3.1. The 0- and 1-codes

The analysis of the D1, T1, and IO verbs is:

- (1) D1: give_v [D1] (x₁) (x₂) (x₃)
- (2) T1: kick_v [T1] (x₁) (x₂)
- (3) IO: pause_v [IO] (x₁)

The number of verbs with a D1, T1, or IO code is very large. I encountered no problems with the examples of IO and T1 verbs that I looked at (but see Chapter 4 on complement-words). The examples of the D1-code, however, made me aware of the fact that LDOCE does not use this code to indicate ditransitive³ verbs only, but all verbs that are followed by two nouns, pronouns, or nounlike expressions. Whereas many one-word verbs which have a D1-code are examples of FG-structures with argument positions for Agent, Goal and Recipient, the multi-word verbs (prepositional, phrasal and phrasal prepositional verbs) are different. Compare:

- (4) I gave her a book
- (5) She ran her fingers through her hair

Both elements have a D1-code and have a 3-place argument structure in the FG lexicon. However, *give* has Recipient function and *run through* has the function Location⁴ on the third argument. It is only because I do not assign semantic functions to the arguments at all that I will not have any problems with the D1-verbs.

For a discussion on L1 and X1, see the sections on L-codes and X-codes.

3.1.3.2. The 2-code

The 2-code is used with I, T, and V-verbs. However, I will not include the I2-verbs in my lexicon, since they are (forms of) the modals and *do*, which should be treated in the grammar rather than be included in the lexicon.

The T2-class consists of 4 verbs, namely *come into*, *come out against*, *flick* and *dare*. None of these verbs will be in our lexicon as (rewritten) T2-verbs. *Come into*, *come out against* and *flick* are not really T2 at all, but T1.⁵ The status of *dare* is not quite clear, but in my view it belongs to the modals and should be treated together with the other modals in the grammar.

The only verb which is a good example of the T2-code is *help*, which, by the way, is included in the Table of Codes at the back of the dictionary (See also Appendix 7) as an example of the T2 code, but does not have this code in the dictionary! Therefore, *help* will not receive the predicate frame associated with the T2-code in my lexicon. For the sake of the argument, however, in the following paragraphs I will assume that it does have this predicate frame .

T2 and V2 can be represented as follows:

- (6) T2: $\text{help}_v [\text{T2}] (x_1) \{(x_2)\} (x_3: [\text{Inf without } to \text{ PRED .. } (x_2) \text{ ..}] (x_3))$
 (7) V2: $\text{see}_v [\text{V2}] (x_1) (x_2) (x_3: [\text{Inf without } to \text{ PRED .. } (x_2) \text{ ..}] (x_3))$

I use curly brackets to indicate that the x_2 with *help* is unspecified and does not occur in an expression like:

- (8) I helped clean the windows

x_2 is the 'understood object' of *help* (or rather the 'understood Goal') and, presumably, the subject of *clean*. In V2-sentences this element does surface, as can be seen in:

- (9) I saw him leave

In this context, x_2 (*him*), which is Goal of *see* and Agent of *leave*, is expressed.

This brings us to the issue of understood elements. It is a well-known fact that the intransitive verb *eat* in 'I'm eating' implies that I am eating **something**. Not all intransitives have an implied Goal, however. CGEL lists some 'pure' intransitive verbs (e.g. *arrive*, *matter*), and some verbs that can be either transitive or intransitive with the same meaning (e.g. *smoke*, *read*).⁶ I could thus represent 'pure' intransitives as I have shown in § 3.1.2. for [I0], and intransitives with an understood element as:

- (10) $\text{eat}_v (x_1) \{(x_2)\}$

where '{ }' surrounds the understood element.

It is probable that all verbs that have both a T1 code and a I0 code in one meaning have an understood element when used intransitively. Unfortunately, I will not be able to investigate this, since in the output of my program it can not be seen whether two codes belong to the same meaning or not.

3.1.3.3. The 3-code

The 3-code proved to be very hard to handle. It is used with the D-, I-, L-, T-, and V-code. The I3-code, to start with, caused particular trouble. The class has to be divided into four groups:

- (11) a. real I3-verbs
- b. verbs with a purpose satellite
- c. verbs indicating a time sequence
- d. modals

An example of a real I3-verb is the verb *yearn*, which should be found in the FG lexicon with either of the following two predicate frames:

- (12) a. $\text{yearn}_v \text{ [I3] } (x_1: [\text{Inf PRED}] (x_1))$
- b. $\text{yearn}_v \text{ [I3] } (x_1) (x_2: [\text{Inf PRED} .. (x_1) ..] (x_2))$

Which of these two constructions covers the I3-code best is not clear. If the first is decided on, then one must explain why the subject of the subclause also functions as the subject of *yearn*. If the second predicate frame is chosen, it must be proved that x_1 is an independent argument. The second argument in this latter construction is not the Goal of *yearn*, because it is not affected by the verb. An example using this verb is:

- (13) They yearned to return home

Other examples are: *ache*, *appear*, *bother*, *endeavour*, *happen*, etc.

The second group of verbs with an I3-code is exemplified by

- (14) She trained to be a doctor

where *to be a doctor* indicates the purpose of the training. It is not obligatory: when the purpose is not indicated, *train* still means 'to be taught a profession, a skill, etc'. The purpose is then implied.

- (15) $\text{train}_v (x_1) \{(x_2)_{\text{Purpose}}\}$

Verbs which have the I3-code, meaning that they may be used in constructions with a purpose satellite, should not be included in the lexicon, because satellites should

not be found in the lexicon. Other examples of such verbs found in LDOCE are: *aim*, *compete*, *cooperate*, *negotiate*, *pay*, *phone*, *race*, and *strive*.

Verbs that are found in the third group of I3 -verbs may occur in structures like:

(16) She lived to be ninety

(17) She awoke to find herself famous

Here it cannot be maintained that *to find herself famous* is the purpose of her awaking and that *to be ninety* is the purpose of her living. However, what is the function of these elements?

The sentences under investigation (16 and 17) are discussed under 'clauses of time' in CGEL. They write: "*to*-infinitive clauses without a subordinator or a subject may have temporal function, expressing the outcome of the situation:

I rushed to the door, only to discover that it was locked and barred (...)

These clauses are restricted to final position, suggesting an analogy between them and result clauses (...), which they resemble in meaning. The sentences can usually be paraphrased by reversing the relationship of subordination (...), and using a when- or after-clause:

When I awoke one morning, I found the house in an uproar (...)

With durative verbs in the matrix clause, the construction expresses duration of time together with outcome:

She lived to be 100"

(p 1079, CGEL).

It is clear that these constructions can, in fact, not be classified under one clause type, but rather belong to two: clauses of time and clauses of result. Not only the verbs coded I3, but also many other verbs can receive such a *to*-infinitive temporal clause. The verb *open*, for instance, which is mentioned in CGEL as an illustration of the type of clause under investigation, is not coded I3 in LDOCE. We can think of many more examples, e.g.:

(18) We waited ten hours to find that they had already left.

These clauses must therefore be treated as satellites and these verbs coded I3 should not be present in the FG lexicon. The reason is, again, that satellites should not be present in the lexicon. A verb such as *live* should not be included in the lexicon on the

basis of its I3-code, but it will be included on the basis of its other codes (e.g. I0). Other examples of such I3-verbs in LDOCE are *blanch*, *blush*, *grow*, and *tremble*.

Members of the fourth and last group of verbs with an I3-code are *be*, *have*, *ought*, and *use*. Examples are:

- (19) You are not to smoke in this room
- (20) Do you have to go now?
- (21) You ought to be ashamed of yourself
- (22) I never used to be interested in ships

These are (semi-)modals that should not be included in the lexicon, but will be treated in the grammar.

I am now in the difficult position that there are quite a few I3-verbs which I should include in my lexicon, but also quite a few which should not be included. My solution to this problem is that I will include all verbs which have an I3-code with predicate-frames as proposed for the "real" I3-verbs, such as *yearn*, above. It should be clear, however, that something is wrong with these entries: the 'illegal' ones will have to be taken out of the lexicon, and for real I3-entries the appropriate predicate frame will have to be selected.

The 3-code is also used with the T- and the V-code. The difference between T3 and V3 lies in the fact that the structure for V3 shows an independent subject in the subclause, whereas for T3 the subject of the subclause is coreferential with the first argument of the main clause. The two are represented as follows:

- (23) T3: want_v [T3] (x₁) (x₂: [Inf PRED.. (x₁) ..] (x₂))
- (24) V3: want_v [V3] (x₁) (x₂: [Inf PRED .. (x₃) ..] (x₂))

Examples are:

- (25) I want to go (T3) (= I want myself to go)
- (26) I want him to go (V3)

This distinction holds nicely for the examples given above, found in the Table of Codes in LDOCE. However, the V3 code is used for a number of different types of verb. The V3 code stands for: 'verb with one object + an infinitive with *to*', or, as CGEL represents it: 'N1 V N2 *to* V N3'. According to CGEL, a distinction must be made between 4 groups:

- (27) a. A *say/tell* group
- b. an *elect/allow* group
- c. an *intend/expect* group
- d. a *want/like* group

Quirk et al. (pp 1216-1220) discuss criteria to keep the four groups apart, among which criteria we find:

- (a') *to* V N3 can be replaced by a finite clause
- (c') change of meaning in passive of N2 *to* V N3
- (e') N2 can become subject of passive.

Many verbs mentioned by CGEL are coded V3 in LDOCE.⁷ Some of them will have to be analysed as 2-place predicates, however, (such as *want* above) while others will be analysed as 3-place ditransitive predicates (e.g. *tell*).

The number of verbs which are ditransitive and have a V3-code is quite large. Thus we find *tell, advise, ask, beseech, bid, cable, cause, recommend* and numerous others. The predicate frame that covers these verbs looks as follows:

(28) $\text{tell}_V [\text{V3}] (x_1) (x_2) (x_3: [\text{Inf PRED} .. (x_2) ..](x_3))$

Unlike the predicate frame for the verb *want*(V3), this predicate frame has three argument positions. There is a relation of coreference between the Recipient (x_2) and the subject of the subclause.

The reason why these verbs have been coded V3 and not D3 is not clear to me. As we saw in §3.1.1, the D3-code is used only once, for the verb *trust*. Obviously, for all ditransitive verbs that have been coded V3, Longman has given priority to structural characteristics over functional characteristics. All constructions that are covered by the pattern 'verb with one object + an infinitive with *to*' get a V3-code, whether they are ditransitive or complex transitive.⁸

This, again, puts me in an awkward position, since I cannot deal with the V3-code properly. The predicate frames that should be used to represent the two classes are so different, and the number of verbs belonging to each of the classes is so large, that I can think of no better strategy than to give them two predicate frames each. I will represent all verbs that have a V3-code as:

- (29) $\text{tell}_V [\text{V3}] (x_1) (x_2: [\text{Inf PRED}](x_2)) \text{ OR}$
 $(x_1) (x_2) (x_3: [\text{Inf PRED} .. (x_2) ..](x_3))$
- (30) $\text{want}_V [\text{V3}] (x_1) (x_2: [\text{Inf PRED}](x_2)) \text{ OR}$
 $(x_1) (x_2) (x_3: [\text{Inf PRED} .. (x_2) ..](x_3))$

These verbs will have to be sorted out by hand later, and the right predicate frame must be selected for each of them. Maybe the criteria given in CGEL will be of help in keeping the groups apart: e.g. verbs that belong to group 1. above are 3-place and those that belong to group 4. are 2-place. Hopefully, the meaning can also be used as an indication

of the class to which a verb belongs (e.g. reportive verbs: 3-place; verbs of feeling or volition: 2-place).⁹

The 3-code is also used with the big letter codes D and L. The D3-code, which is used only for *trust*, will be rewritten as is shown in § 3.1.2. and the L3-code will be discussed in section 3.1.6. together with the other L-codes.

3.1.3.4. The 4-code

The number code 4 is used with big letter codes I, X (*save*), L, T, and V. The I4-code is used for the verbs *be*, *bu(r)st out*, *come*, *forbear*, *go*, *go on*, *write*, and *write back*. With *be*, I4 is used to indicate its use as an auxiliary of the progressive: this should not be included in the lexicon, but should rather be treated in the grammar. The verbs *bu(r)st out* ('to begin suddenly (to use the voice without speaking)') and *go on* ('to continue without stopping, or after a stop') have an aspectual meaning. *Forbear*, which has a negative aspectual meaning ('to make no attempt to do something that one has the right to do ...') means that the action did not take place. *Go*, which is used in phrases like *go shopping*, might also be said to be an aspectual verb. It has almost no meaning except 'to do (an activity)'. To the question 'What did you do yesterday?' one might answer either: 'We went shopping' or 'We shopped' where the first answer is the most frequently used one.

Aspectual verbs involve the beginning/continuation/ending of an action. Verbs indicating aspect are part of the grammar, and are not in the lexicon. How aspect is handled in the grammar is something which does not concern me here. I think that a very sophisticated and elaborate system of rules is needed to insert the right aspectual verbs in the right cases. The system would have to be able to handle all aspectual verbs that are found in LDOCE (See also the discussion in the T4-code).¹⁰

The situation is different with the verbs *come*, *write*, and *write back*. These verbs say nothing about the action expressed in the *-ing*-form, as can be seen in the following examples:

- (31) I came running
- (32) I wrote (back) asking them to come

Rather, the I4-verb and the *-ing*-form express simultaneous actions/events. The question might be asked whether the 4-code is really necessary here. The *-ing*-phrases are optional: they can be left out without changing the meaning of *came* and *wrote*, respectively. Moreover, each verbal construction may occur with such an *-ing*-form following it, not only verbs coded I4.

- (33) She fell down crying
- (34) I gave her a chocolate thinking it would soothe her
- (35) The man hit her, thinking he was attacked

For the verbs *come*, *write* and *write back*, the I4-code is not necessary. The I0-code, which they all have, is sufficient.

The T4-code can be divided into three main groups (CGEL p 1189):

- (36) a. emotive verbs. E.g. *enjoy*, *resent*, *dislike*
- b. aspectual verbs. E.g. *start*, *stop*, *get*
- c. miscellaneous. E.g. *admit*, *risk*, *remember*

The aspectual verbs ought to be handled by the grammar, rather than the lexicon (see above). The *-ing*-phrases of the verbs in the other two classes can be analysed as nominalized clauses (Cf Mackenzie, 1983). If we treat them as such, these verbs are simply transitive, but with a nominalized clause as goal.

The information that the second argument is a nominalized clause distinguishes these verbs from T1 verbs like *fold*: $\text{fold}_V(x_1)(x_2)$. T1 verbs may not have a nominalized clause as second argument if they do not also have code T4:

- (37) I folded the piece of paper
- (38) *I folded making a small parcel
- (39) I hated the play
- (40) I hated playing that part

Most, but not all T4 verbs also have other T-codes. Two (prepositional) verbs have only code T4, namely *kick against* and *kick at* (both with the sense: 'oppose or dislike (doing something)').

My proposal that we are dealing with nominalized clauses in the above cases is supported by the existence of verbs that have code V4: 'verb + object+ *-ing* form'. Besides its T4 code, *hate* also has a V4-code:

- (41) He hates people asking him for money

In these constructions the subject of the nominalized clause surfaces, while it did not surface in case of the T4-verbs. The difference between the two structures can be seen in their predicate frames:

- (42) T4: $\text{hate}_V[\text{T4}](x_1)(x_2: [\text{NOMINALIZED PRED} ..(x_1) ..](x_2))$
- (43) V4: $\text{hate}_V[\text{V4}](x_1)(x_2: [\text{NOMINALIZED PRED} ..(x_3) ..](x_2))$

I use x_1 in the subclause of T4 to indicate that the subject of the subclause is coreferential with the first argument of the main clause. In the predicate frame of the V4-verb, the subject of the subclause is not coreferential with another argument in the structure, and therefore has an argument, (x_3) , which does not occur elsewhere in the frame.

Another point in favour of my proposal that we are dealing with nominalized clauses is the existence of verbs that allow a genitive pronoun as subject of the *-ing*-clause, which are discussed in CGEL (p. 1194):

(44) I hate his driving my car

The verb *hate* does not have a separate code for this type of construction in LDOCE. The possibility of such a construction might be assumed to follow from the fact that the verb has a V4-code, which also takes care of the following construction:

(45) I hate him driving my car

However, not all V4-coded verbs allow the genitive:

(46) I caught him reading

(47) *I caught his reading

In CGEL, the verbs of this class are divided into 3 groups:

- (48) a. verbs of perception
 b. verbs of encounter
 c. two verbs of coercive meaning: *have* and *get* (CGEL p 1206)

I will take it that the verbs *have* and *get* do not belong in the lexicon, but in a grammatical procedure for causing (Cf Section on code V8).

Verbs from LDOCE which belong to group 1) are *see* and *smell*; verbs belonging to group 2) are *catch* and *find*. The verb *hate*, however, together with many other verbs with a V4-code in LDOCE, does not fit in any of these groups. Incidentally, all these 'extra' V4-verbs allow a genitive pronoun, but none of the other V4-verbs do:

(49) I hate him/his driving my car

(50) I caught him/*his reading

(51) I saw him/*his leaving

The difference is related to the fact that in (49) *him* is only the subject of the *-ing*-clause, whereas in (50) and (51) *him* is a separate argument in the frame of *catch* and *see*, respectively.

(52) hate_v [V4] (x₁) (x₂: [NOMINALIZED PRED] (x₂))

(53) see_v [V4] (x₁) (x₂) (x₃: [NOMINALIZED PRED .. (x₂) ..] (x₃))

Just as with the V3-code, we can see that two constructions are involved in the treatment of the V4-code, both of which are used for a fair number of verbs. Since I know of no way of distinguishing these, I will include V4-entries as follows:

- (54) hate_v [V4] (x₁) (x₂: [NOMINALIZED PRED] (x₂)) OR
 (x₁) (x₂) (x₃: [NOMINALIZED PRED .. (x₂) ..] (x₃))
 (55) catch_v [V4] (x₁) (x₂: [NOMINALIZED PRED] (x₂)) OR
 (x₁) (x₂) (x₃: [NOMINALIZED PRED .. (x₂) ..] (x₃))

The correct predicate frames will have to be selected manually, later.

So far, I have only spoken of one-word verbs having a V4-code. However, there are many multi-verbs that allow a subordinate *-ing*-clause. Of all 114 verbs that have a V4-code in LDOCE, 77 are multi-word verbs (70 prepositional, 6 phrasal prepositional and 1 phrasal) and 37 are one-word verbs. Of the 237 T4-verbs, 119 are multi-word verbs (88 prepositional verbs, 14 phrasal prepositional verbs and 17 phrasal verbs) and 118 are one-word verbs. It is very likely that the *-ing*-form with many of these verbs is due to the preposition. Although each one-word verb with a V4-code will get two predicate frames, I will give one predicate frame to each multi-word V4 and V4b verb, as follows:

- (56) approve of_J [V4] (x₁) (x₂: [NOMINALIZED PRED](x₂))
 (57) talk into_J [V4b] (x₁) (x₂)
 (x₃: [NOMINALIZED PRED ..(x₂) ..](x₃))

The difference between these structures can be seen in the following examples:

- (58) I don't approve of silly people wasting time
 (59) My wife talked me into buying her a new car

Multi-word verbs with a T4 code will get, like all other T4-verbs, frames like:

- (60) reckon on_J [T4] (x₁) (x₂: [NOMINALIZED PRED ..(x₁) ..](x₂))

An example showing this construction is:

- (61) You can't reckon on seeing him

Finally, the 4-code is used for some linking verbs, which will be discussed in section 3.1.6, and for the verb *save* which is coded X4. Actually, *save* should be D4, because there is a Recipient involved, as can be seen in the example from LDOCE, given in section 3.1.1., which I repeat here:

- (62) It'll save me going into town

The frame for this verb is:

- (63) save_v [D4] (x₁) (x₂) (x₃: [NOMINALIZED PRED ..(x₂) ..] (x₃))

3.1.3.5. The 5-code

D5 and T5 do not cause any problems. The L5-code is discussed together with the other L-codes. I5 should be divided into two groups: those verbs which are real I5 (the *it*-constructions, which can be found by looking for the complement words 'it+' and 'it-'), and those which should in fact be T5 (*grizzle*, *vote*, etc.)

- (64) D5: warn_v [D5] (x₁) (x₂) (sub x₃: [PRED] (x₃))
 (65) T5: know_v [T5] (x₁) (sub x₂: [PRED] (x₂))
 (66) real I5: appear_v [I5] (sub x₁: [PRED] (x₁))

The [a] and [c] possibly following the [5]-code indicate variations on the [5]-theme and do not cause any problems.

3.1.3.6. The 6-code

The 6-code is used with big letter codes D, I, L and T. For a discussion of the L6-code, see the section on L-codes (§ 3.1.6.). The meaning of the I6-code can be indicated by [-FACT], as opposed to [+FACT], which is used for all predicates that have an I5-code:

- (67) I6: appear_v (sub x₁: [PRED (-FACT)] (x₁))
 (68) I5: come about_F (sub x₁: [PRED (+FACT)] (x₁))

Appear [+FACT] would give rise to sentences like:

- (69) It appears that she will win

[-FACT] leads to:

- (70) It appears as if she will win

Appear and *seem* are the only verbs having an I6 code.

When used with the D, and T-codes, the 6-code indicates that the predicate is followed by a *wh*-word with either a finite clause ([6a]) or an infinitive clause ([6b]):

- (71) Tell me where to go (D6b)
 (72) He anticipated what would happen (T6a)

Not all constructions that allow a *wh*-word in this position and are (di)transitive have a T/D6-code:

- (73) They gave her what she wanted

The verb *give* has codes D1, T1, L9, V3, X9 and I0 and neither a D6 nor a T6 code. Apparently, the T/D6-code means that the *wh*-form in the subordinate clause may, but need not necessarily be the goal of the verb in the subclause, as it is in (73). Compare:

- (74) Tell me what to do
 (75) *They gave her who wanted it

We can see here that with *tell*, which has code D6b, the *wh*-form may both be the goal of the verb in the subclause (*do*) and something else. With *give*, however, the *wh*-form may be the goal of the verb in the subclause (*want*), but it may not have any other function.

I will represent the 6-code as:

- (76) D6b: tell_v [D6b] (x₁) (x₂) (*wh* x₃: [PRED ..(x₂) ..] (x₃))
 (77) T6a: anticipate_v [T6a] (x₁) (*wh* x₂: [PRED] (x₂))

Note that in the structures in which the D6b-code is used, the subject of the *wh*-clause must be coreferential with the Recipient (x₂) of the main clause, and when T6b is found the subject of the *wh*-clause must be coreferential with the subject of the main clause (in the next examples I will use an *i*-subscript when elements are coreferential) :

- (78) Tell me what to do (= Tell me_i what I_i shall do) ([D6b])
 (79) We haven't fixed where to stay, yet
 (= We_i haven't fixed where we_i will stay) ([T6b])

3.1.3.7. The 7-code

For a discussion of the 7-code see the sections on X-codes and L-codes.

3.1.3.8. The 8-code

The code I8 is used for *be* and *have* as auxiliaries of the passive and the perfect, respectively. They should not be in my lexicon, but should be treated in the grammar.

The V8-code is used for two causatives: *get* and *have*, two volitional verbs: *want*, *need* and one perceptual verb: *see*. In fact, CGEL lists some more verbs of the last two categories, which, however, do not have V8 in LDOCE. In general, the complementation of these verbs is semantically equivalent to one with an infinitive form of *be*:

- (80) I want my room cleaned
 (81) I want my room to be cleaned

We may want these verbs dealt with in the grammar together with the causatives or we may want to describe the V8-verbs as structures like:

- (82) want_v [V8] (x₁) (x₂: [Perf PRED] (x₂))

Causative formation is a very general process. It is achieved in many languages by means of adding a causative affix to a verb, rather than by adding a word like *make* or

cause as in English. However, I know of no affixes for expressing *need*, *see* and *want*, so these will not be handled in the grammar.

The verbs *have* and *get*, however, are so close in meaning to *make*, and *cause*, which are involved in causative formation in English (Cf Dik, 1980: Ch 3), that I will leave them to be handled together with *make* and *cause* in the grammar of causative formation. I think that the V4-verbs *have* and *get* also belong there. Compare:

- (83) V4: I'll get the car going
I can't have you running up and down all day long
(84) V8: He got it done
To have a house built

Although I think that the predicate frame shown for *want* above covers the use of *need*, *want* and *see*, I will not include the V8-verbs in my lexicon. It is better to look more carefully at these verbs, and to compile a list of all verbs that allow the V8-pattern. This list would also include the verbs mentioned in CGEL that do not have V8 in LDOCE.

3.1.3.9. The 9-code

The 9-code is used with big letter codes X and L for verbs. They will be treated in the sections that deal with these codes (§3.1.5. and 3.1.6., respectively).

3.1.4. Some remarks on the V-class

The V-class resembles what in CGEL are called complex transitive verbs. Many verbs listed as complex transitives in CGEL have a V-code in LDOCE. According to CGEL the defining property of complex transitives is the 'divisibility of the semantically clausal construction following the verb' (CGEL 1195). *See* is a good example of a complex transitive verb, since the subject of its subclause can be set apart from the subclause when it becomes the subject of the matrix clause in passivization:

- (85) I saw her leaving
(86) She was seen leaving

However, with many verbs that have a V-code in LDOCE the subclause cannot be likewise divided.

- The V2-verbs can be passivized, but only when the infinitive marker *to* is inserted:

- (87) I saw her leave
(88) *She was seen leave
She was seen to leave

- Some of the V3-verbs can and others cannot be divided:

- (89) I want you to leave
(90) *You are wanted to leave

- (91) I told her to go
 (92) She was told to go

- Some of the V4 can and others cannot be divided:

- (93) *People are hated asking him for money
 (94) She was caught reading
 (95) ?She was watched leaving

- Only one of the V8-verbs can be divided:

- (96) The man was seen shot
 (97) ??It is needed finished
 (98) ??The room was wanted cleaned
 (99) *The watch was had/got repaired

3.1.5. The X-code

The X-code is used to indicate complex transitives in which the element following the 'object' is non-verbal. This category is complementary with the V-class. All numbers which occur with X are not found with V, and vice versa. The reason for this is simply that the definition of the V-class is 'v+obj+verb form', while that of the X-class is 'v+obj+something else', so the definitions are complementary. The numbers 1, 7, and 9, which are used with X, indicate that the V+object is followed by one or more nouns or pronouns (1), by an adjective (7), or that it needs a descriptive word or phrase (9). With X, therefore, all numbers indicate non-verbal forms, which is exactly the definition of X, whereas the numbers used with the V-class all indicate that a word is followed by a verb form, which is the definition of the V-class.

The words that have an X-code in LDOCE are an interesting group. This group contains verbs like *consider* (X1 and X7), and *appoint* (X1). I had particular difficulty in describing the X1-verbs. There are reasons to believe, for instance, that *consider* is a 2-place predicate, and that *appoint* is a 3-place predicate.

I have tried to find ways to keep the 2- and 3-place predicates with an X-code apart. I first checked if I could make all X1 verbs 2-place, and all X7 verbs 3-place, for instance. However, that this would not work for the *consider*-example given above will be clear: there is no difference in valency of the verb *consider* in the next two examples:

- (100) I consider him the best programmer
 (101) I consider him crazy

Both are, in my opinion 2-place (See De Groot et al., 1986). I then tried to see if all 2-place X-predicates could be kept apart from 3-place X-predicates by testing for the presence of the complement words *to be*. The argumentation for this test goes as follows: If a verb can have *to be* in its complementation, then the two elements which this copula

connects will probably form a sub-clause, and together they form one argument. In this way we can keep apart the following two sentences:

- (102) I consider him (to be) a fool/crazy
 (103) I appoint him (*to be) chairman of the committee

Apparently, my intuition on the impossibility of *to be* in the second example was wrong, since LDOCE does list *to be* as a complement word for the verb *appoint*.

The verbs that have only X1 and not X7 do, however, form a semantic group: they are all verbs that classify or select and include verbs of naming (e.g. *name, baptize*), verbs of appointing (e.g. *appoint, constitute*), and verbs of choosing or identifying (e.g. *choose, mistake for, suspect, know*). For a list of X1-but-not-X7-verbs see Appendix 5. Although the verb *appoint* is quite probably 3-place, in De Groot et al. (1986), it was found that we could say nothing conclusive about verbs of naming. Note, by the way, that the verb *call*, which belongs to the group of verbs of naming, is coded X1 and X7. This shows that there may be verbs that belong to this semantic class which are not coded X1-and-not-X7.

We conclude that the X1-code is ambiguous. Verbs that have an X1-code will be treated as follows: those that also have an X7-code will get one single predicate frame in the lexicon, but the verbs that have X1 and not X7 will get two predicate frames:

- (104) X1-and-not-X7: appoint_v [X1] (x_1) (x_2) (x_3) OR
 (x_1) (x_2 : [(Inf) PRED] (x_2))
 (105) X1-and-x7: consider_v (x_1) (x_2 : [(Inf) PRED] (x_2))

For all X1-verbs that have two predicate frames in the lexicon, the correct frame will have to be selected.

The X7 verbs are all 2-place predicates with predicate frames like the following:

- (106) zip_v (x_1) (x_2 : [PRED_{adjectival}] (x_2))

One might wonder whether code X7 is not simply the same as code T1, where the second argument is modified by an adjective following it. Almost all X7 verbs also have code T1. Exceptions are: *prize/prise* ('to raise, move, lift or break with a tool or metal bar. PRY), *rate, rave, tumble, zip*. That X7 is not a variant of T1 becomes clear when we consider the following examples:

- (107) He bought the car cheap
 (108) He bought the cheap car

Sentence (108), where *cheap* modifies *car*, means something totally different from (107), where *cheap* does not modify *car*. In (107) the car may have been expensive, but he bought it for a low price. *Cheap* tells us something about *buy* rather than about *the car*. In (108), however, *the car* was *cheap* but we are not told whether he paid much or little for it.

I will finish this section on X-codes with the X9-code. This code is used very frequently in LDOCE. As X is used for the pattern 'v with one object + something else' and 9 stands for 'needs a descriptive word or phrase', one may expect to find various types of verbs in this class. Examples of verbs with an X9-code are:

- (109) *drop* [X9] Drop me at the corner
 (110) *ship* [X9 (OFF)] He joined the army and was shipped off to training camp
 (111) *scale* [X9, esp. UP, DOWN] scale up/down taxes

Many of these verbs can be described by the pattern for *put*

- (112) $put_v(x_1)(x_2)(x_3)$

where the third argument has Loc function. However, other functions are also possible, e.g.:

- (113) Direction: *ship* - see example above
 cram - to cram people into a railway carriage
 (114) Source: *suspend* - to suspend a rope from a tree
 take - you took your article directly from my book

3.1.6. The L-codes: linking verbs

The L-code is used with number codes 1, 3, 4, 5, 6, 7, 8 and 9. The L3-, L4-, L5-, L6-, and L8-codes are used only for a very small group of verbs:

L3: *be, incline, turn out*:

- (115) The difficulty is to know what to do
 (116) I incline to take the opposite view
 (117) He turned out to live in Hastings

L4: *be, come across as, end up, wind up*

- (118) The trouble is knowing what to do
 (119) He came across as being quite a nice person really
 (120) She ended up dancing on the table
 (121) He wound up feeling ashamed of himself

L5: *be, go*

- (122) The trouble is that you know too much, my dear Bond
 (123) The story goes that he was murdered

L6: *be*

- (124) It's as if we'd never even started

L8: *get*

- (125) He got trapped

One might prefer to view the verb *incline* as a sort of modal verb rather than as a linking verb.

The number of verbs having one of these codes is small, but the remaining L-codes are given for larger groups of verbs: LDOCE has 66 verbs with code L1 (Note that CGEL lists 14 verbs that may be used with a nominal complement (= LDOCE L1)!). The number of verbs that may be used with an adjectival complement (= LDOCE L7) is only slightly smaller than the number of L7-verbs in LDOCE: 55. The patterns that L1 and L7 stand for are exemplified by:

- (126) She became queen (L1)
 (127) She became famous. (L7)

I think it is difficult to decide whether all the linking verbs that we have seen so far are to be treated in the grammar or in the lexicon of Functional Grammar. The disadvantage of treating them in the grammar is that the grammar will then have to contain a very sophisticated system for deciding which of the linking verbs is used in which context. The system might have to make use of very specific information given for each linking verb. The more specific information is needed, the less general the system can be.

However, grammatical treatment could also have its advantages. The difference and similarity between the verbs *be* and *become* have been pointed out in De Groot et al. (1986). Their similarity lies in the fact that they can be used in comparable contexts and treated by one mechanism. The difference between these two verbs can be expressed in just one component in the rules in this mechanism (namely + or - DYNAMIC) (N.B. LDOCE does not distinguish current and resulting copulas as CGEL does). If these two verbs can be treated by one and the same mechanism, one might want to try and include other linking verbs in this mechanism as well. Although this matter clearly needs more attention, I will not pursue it here. The linking verbs will, for the time being, not be included in the lexicon (except verbs with code L9, most of which are not linking verbs anyway).

The class of linking verbs also contains verbs like *cost*, *weigh*, etc., used in contexts like:

- (128) The best goods usually cost most
 (129) It weighs six pounds

One might make the mistake of thinking that *cost* and *weigh* are transitive verbs rather than linking verbs. However, the difference with transitive verbs becomes clear when we compare (129) with (130):

- (130) He weighed the cheese

Whereas in (130) *the cheese* is the Goal of *weighed*, in (129) *six pounds* cannot be the Goal. *Weigh* in (129) gives a property of *it*, namely that its weight is six pounds.

I will now go on to discuss the verbs which have code L9. These verbs are numerous and include *live*, which can be used in sentences like:

(131) He lives in London

However, used in this context *live* should clearly have code I9 and not L9, because it is an intransitive, rather than a linking verb. In fact, *live* is not the only verb which should have had an I9 instead of an L9-code. Other examples are: *fly*, *hang out*, *hover*, *land*, *nestle*, and *sit*. There are only a few verbs that are real L9, such as *turn*, *be*, *grow*, etc.

A further problem with the L9-code is that what for *live* must be a Loc argument may for other verbs be an argument with some other semantic function. Examples are:

(132) Direction: *flock* - In the 19th century people flocked to the cities

pile - They all piled into the room

(133) Source: *rise* - Christ rose from the dead

I will give to all entries with an L9-code a predicate frame like that for *live*, with no semantic function indicated on the last argument:

(134) $\text{live}_v(x_1)(x_2)$

3.2. The noun codes

3.2.1. Nominal Predicates

As might be expected, grammatical number codes indicating complementation are used far less frequently for nouns than they are for verbs. For verbs Longman uses the whole range of number codes from 0 to 9. These numbers are attached to Big letter codes which they, as it were, subdivide. For instance, the T in code T5 indicates that the verb for which it is used is transitive, or, in FG-terms, that it has two arguments; the 5 in the T5 code indicates that the second argument has a special form, namely that of a finite subclause introduced by *that*.

With nouns, only the numbers 3, 4, 5, 6, and 9 are used. Number 4 is not mentioned as a possibility in the Table of Codes. (The 0 used with verbs is basically the same as 'no number' for nouns). This indicates that the range of complementation types is smaller than for verbs. Moreover, the numbers do not subdivide the grammatical codes with nouns, rather they give a different type of information. Big letter codes for nouns give information about:

- (135) a. countability (C, U, GC, GU, S, P)
 b. position with respect to another noun (A, E)
 c. function (N)

The numbers give information on complementation. The reason why the relation between big letter code and number code is different for nouns than it is for verbs is clear: verbs allow a much wider range of complementation types, and complementation (other than I0, which means no complementation) is very frequent, whereas most nouns do not have complementation at all.

The distribution of noun codes (except the R-code, see §3.2.2.) can be seen in the following matrix. Each number in the matrix indicates how many entries in LDOCE have at least one occurrence of a grammatical code. (For an overview of the grammatical codes in LDOCE see Appendix 7).

	A	C	E	GC	GU	N	P	S	U
- 11	460	18972	21	155	144	494	860	909	8105
3		57					1	13	36
4		1							
5		77						10	30
6		3					1	2	3
9	2	311		4	4	28	34	114	165

Nouns with codes which do not occur in the Table of Codes, but which do occur in LDOCE are:

A9: *octane, rate*

C4: *rider*

GC9: *class, contingent, corps, fraternity*

N9: 28 nouns, e.g. *bastard, excellency, lordship*; these are mostly verbs which can be used with complement word *you+*

P6b: *directions*

S6: *moot point, moot question*

U6: *conception, doubt, uncertainty*

I can see nothing wrong with the codes used for these words, so I assume Longman has simply forgotten to put them in the Table of codes. Examples illustrating the use of these codes (except A9, GC9, and N9) are:

- (136) [C4] He added a rider advising that the lake should be filled in
 (137) [P6b] His directions how to get to the farm were quite clear
 (138) [S6] It was a moot point/moot question where to go
 (139) [U6] I have no conception why you left home
 (140) „ There's no doubt why he did it
 (141) „ There's some uncertainty whether she really knows this

The noun codes that have a big letter code but no number code are rewritten as 1-place structures, with the big letter code put in a fixed position (following the lemma and its wordclass) and between square brackets, as follows:

(142) jury_N [GC] (x₁)

The predicate frames of nouns that have a number code (except 9) are rewritten as follows:

(143) 3: attempt_N [C3] (x₁) (x₂: [Inf PRED] (x₂))
 4: rider_N [C4] (x₁) (x₂: [NOMINALIZED PRED] (x₂))
 5: argument_N [C5] (x₁) (sub x₂: [PRED] (x₂))
 5a: impression_N [C5a] (x₁) ((sub) x₂: [PRED] (x₂))
 5c: desire_N [C5c] (x₁) (sub x₂: [SUBJUNCTIVE PRED] (x₂))
 6a: doubt_N [U6a] (x₁) (wh x₂: [PRED] (x₂))
 6b: directions_N [P6b] (x₁) (wh x₂: [Inf PRED] (x₂))

Number code 9 will not be used to determine the argument structure. It is used to indicate that the predicate 'needs a descriptive word or phrase', as found with the following underlined items: a tennis buff, his posterity, powers of memory, the third International, a fine intelligence, jurisdiction over us all. I find these too difficult to represent formally, so I will simply represent codes which include a 9 as:

(144) powers_N [P9] (x₁)

Many nouns which have a number code can be argued to be derived from verbs that have a similar complementation type by a rule of predicate formation (nominalization). For instance, the noun *motivation*, which, among others, has code [C3], may have been derived from the verb *motivate*, which has code [V3]:

(145) a. motivate_V (x₁: I(x₁))_{AgSubj} (x₂: he(x₂))_{GoObj}
 (x₃: [Inf go_V (x₂)](x₃))
 b. (d1x_i: [motivat-ion_N (x_j: I(x_j))_{AgPoss} (x_k: he (x_k))_{GoPoss}
 (x₁: [Inf go_V (x_k)](x₁))] (x_i))₀
 c. (d1x_i: [motivat-ion_N (x_j: he(x_j))_{GoPoss}
 (x_k: [Inf go_V (x_j)_{Ag}](x_k))] (x_i))₀

The expressions of these structures are:

(146) I motivated him to go
 (147) My motivation of him to go
 (148) His motivation to go

As we can see in (147), the Subject Agent (*I*) of *motivate* in (146) may become the Possessor Agent of the nominalized construction and is expressed pre-nominally (*My*).

The Goal of *motivate*, *him*, is expressed as a post-nominal Possessor (*of him*). In (148), however, the Goal of *motivate* in (146) is expressed as a pre-nominal Possessor Agent (*His*), and the original Subject of the verb *motivate* is not expressed. In (147) we are dealing with a nominalization of (146), but in (148), the structure underlying (146) has undergone Valency Reduction before it was nominalized (Cf J.L. Mackenzie, 1985).

Some examples of nouns with a number code from LDOCE suggest that nouns may be derived from adjectives as well. Examples are: *readiness*, *necessity*, and *anxiety*. If one wants to set up any rule of nominalization, however, whether it has an adjectival predicate or a verbal predicate as input, it will first have to be established whether the rule is synchronically productive or not.

If it is decided that some nouns with a number code in LDOCE can be regarded as nominalizations, they will have to be removed from the lexicon manually. I will include all nouns with number codes in my lexicon, because:

- within the scope of this paper, it is impossible for me to decide in each case whether a noun has been formed by a synchronically productive rule or not
- there are also nouns with a number code which cannot be regarded as nominalizations, because no corresponding verb or adjective exists. Examples of such nouns are: *idea*, *effort*, *instinct*, *notion*, *theory*, *carte blanche*, *evidence*, *yen*, and *power*.

Note, finally, that LDOCE does not give any complementation codes for relational nouns, like words denoting family relations. Although the word *brother* could be argued to be a two-place nominal predicate (one must always be a brother of another person) this word will appear as a one-place predicate in my FG lexicon on the basis of the codes given in LDOCE: C, N, and A.

3.2.2. Terms

Dik (1980) mentions that pronouns and proper names will be included in the lexicon as basic terms. What, exactly, are the properties that these elements share, and which make them different from nominal predicates? Some characteristics could be:

- 1) Pronouns and proper names have unique reference when used in context;
- 2) Pronouns and proper names can be inserted in the predicate frame without having to go through the process of term-formation.

The first characteristic does not distinguish between basic terms and basic nominal predicates, since all nominals which are used definitely (e.g. *that apple over there*) have unique reference in context. The second characteristic is a better criterion.

In LDOCE, we find pronouns, but not proper nouns. There is also a category, indicated by the grammatical code R, which is used for each 'noun that is a name or namelike'. This category is used for

- the months of the year (February, December)
- the signs of the zodiac (Taurus, Gemini)
- languages (Afrikaans, English)
- North, South
- the Bible, the Cross
- letters of the Greek alphabet: beta, iota,
- etc

Entries which have grammatical code R are either always used with *the*, or they are never used with *the*. These entries are, I think, good examples of basic terms.

Unfortunately, the R-code is also used for various other words that have unique reference for the language user in context, like: the *city hall*, the *underground*, etc. Although these problem-cases exist, I will use nouns with an R-code, together with pronouns, as basic terms.

The number of nouns in LDOCE which have an R-code is as follows:

	R
-	1071
3	2
5	1
9	30

Nouns with codes which do not occur in the Table of Codes, but which do occur in LDOCE are :

R3: *mood, wherewithal*

R5: *message*

Examples:

(149) [R3] She was in no mood to dance

(150) „ I'd like a new car but I lack the wherewithal to pay for it

(151) [R5] Christ brought the message that God loved the world

The R9-code is used with: *best, bug, cost of living, Creator, earth, eve, Expo, front, like, Mach, marine, middle name, missus, missis, operation, Place, plate, Point, Proctor, Reich, Republic, riding, Road, Row, small, talk, time, trade, vote, world*.

I will not include the R-codes which are used with a number code in my lexicon. The other R-codes will be included as in the following term structure (the i/d is included because Longman does not indicate whether a term should be definite or indefinite):

(152) (i/dnx₁: February_N [R] (x₁))

3.3. The Adjective Codes

Adjectives, like nouns, are usually one-place predicates. All adjective codes, except GU, are used to indicate position with respect to other elements:

A = used before a noun	(attributive use)
E = used after a noun	(attributive use)
B = used before or after a noun (= A + F)	(attributive and predicative use)
F = used after a verb	(predicative use)

In fact, it is questionable whether the GU-code should be used with adjectives at all. This code is used with adjectives like *above-mentioned* to indicate that they can be used in phrases like *the above-mentioned*. I think these cases should be handled in a rule of predicate formation that creates nouns on the basis of adjectives.

The following table gives the numbers of occurrences of combinations of number and Big letter codes (Cf Appendix 7 for a copy of the LDOCE Table of Codes). Each cell in this table indicates the number of adjective entries that have at least one occurrence of the combination of grammatical codes shown on the vertical and horizontal axis.

	A	B	E	F	GU
-	1139	6733	89	481	10
3		14		83	
4		1		2	
5		5		32	
6				8	
7				1	
9	5	11		20	

Codes which are used and which are not accounted for in the Table of Codes on the back cover of the printed version of LDOCE are:

- A9: colonial, formula, Hono(u)rable, Hon.
- B4: busy
- B5: anxious, characteristic, conceivable, fearful, thankful
- F4: bent (up)on
- F7: scared

Again, I will not concern myself with the 9-codes. These include the underlined adjectives in the following contexts: politically aware, the Honourable Glencora Smith-Fortescue, located in Florida, etc.

Neither the B4-code nor the F4-code is in the Table of codes. Examples showing the use of these codes are:

- (153) B4: He is busy writing (B4)
- (154) F4: Jim seems bent on becoming a musician (F4)

The *-ing*-form may not follow just any adjective (at least, when the *-ing*-form modifies the adjective), as can be seen in the following examples:

- (155) *She was free writing
 (156) *It was difficult skiing
 (157) *The sea was calm crossing

I think that the B4-code for *busy* is justified, and I will use it in my lexicon. Matters seem to be different with *bent on*, however. The *-ing*-form is triggered by the presence of the preposition *on*. However, since the combination *bent on* has the wordclass adjective, and since it is in no way shown that this adjective consists of a participle and a preposition, I cannot but agree with the use of the 4-code here. The fact that *bent on* and *bent upon* should be the only examples of such prepositional adjectives can be explained by the fact that the prepositional verb *bend (up)on* does not exist. Whenever such a verb does exist, the adjective is not included in the lexicon, but the verb has grammatical code Wv5 ('used in *-ed*-form as adjective').

I have difficulties with the 5-code used with adjectives. The sentences which illustrate this code in combination with the big letter code B, a combination which is not allowed in the table of codes, are:

- (158) He was anxious that they should have all they want
 (159) It was characteristic that he behaved badly
 (160) It is conceivable that my wife missed the train, but it's very unlikely
 (161) He was fearful that she should be angry
 (162) You should be thankful that you're alive

The two examples with *It* as subject, (159) and (160) should simply be considered as one-place predicates. The other three adjectives, however, seem good examples of adjectives which can have a subclause. Since the correctly coded adjectives outnumber the incorrect ones, I will choose to code all as shown below.

The code F7, finally, seems fully justified in case of the adjective *scared*, so I will retain it in my lexicon. An example showing the use of this code is:

- (163) I'm scared stiff.

The numbers in the adjective codes will be rewritten according to the following rules:

- (164) 3: anxious_A [B3] (x₁) (x₂: [Inf PRED .. (x₁) ..] (x₂))
 (165) 4: bent on_A [F4] (x₁) (x₂: [NOMINALIZED PRED .. (x₁) ..] (x₂))
 (166) 5: insistent_A [F5] (x₁) (sub x₂: [PRED] (x₂))
 (167) 5a: glad_A [F5a] (x₁) ((sub) x₂: [PRED] (x₂))

- (168) 5c: fearful_A [B5c] (x₁) (x₂: [SUBJUNCTIVE PRED] (x₂))
 (169) 6a: aware_A [F6a] (x₁) (wh x₂: [PRED] (x₂))
 (170) 6b: sure_A [F6b] (x₁) (wh x₂: [Inf PRED .. (x₁) ..] (x₂))
 (171) 7: scared_A [F7] (x₁) (x₂: [PRED_A (x₁)] (x₂))

3.4. The Grammatical Codes: Conclusion

3.4.1. Types of entries in the FG lexicon

In an FG lexicon, we find basic terms and basic predicates. Basic predicates can be nominal, adjectival, or verbal. Nominal predicates have been derived from entries in LDOCE with wordclass n (except those that have an R-code); adjectival predicates have been derived from LDOCE-entries with wordclass adj, and verbal predicates have been derived from LDOCE entries with wordclass v, v adv, v prep, and v adv prep.

The typical functions of verbs, adjectives and nouns in an FG lexicon are:¹²

verb	-> predicate
adjective	-> predicate
	-> restrictor
noun	-> head of a term

Verbs function as predicates. Adjectives may be used as predicates (predicative use), or as restrictors (attributive use). Nouns are typically used in term formation to become head of a term. It is questionable whether, in English, bare nominals may be used as predicates. When nominals are used as predicates in English, they have undergone TERM FORMATION and TERM PREDICATE FORMATION.

These typical functions of verbs, nouns and adjectives are reflected in the grammatical codes that these entries have in LDOCE. In the table showing the frequencies of noun codes in LDOCE, it can be seen that the vast majority of the noun codes indicate countability: the C, GC, GU, S, P, and U-codes together constitute 96.8% of all noun-codes. The information on countability is related to the term operator for number of the term of which the noun is the head (Cf Dik, to appear).

All adjective codes (except the GU-code, the value of which is questionable in adjective coding, anyway) bear on the function of adjective: thus A and E indicate adjectives which are used as restrictors (the difference is that an adjective with an A-code comes before the noun, and an E-adjective comes after the noun); F indicates the adjectives that are used as predicates; and B indicates adjectives that may both be used as a restrictor, and as a predicate.

All verb codes are used to give information on the argument structure of the verbal predicate. In the same way, the subcategorization by number of adjectives

indicates the argument structure connected with adjectival predicates. The number codes used with nouns, finally, give information on the internal structure of the term of which the noun becomes the head.

Are the codes indicating argument structure for adjectives and verbs, which can both be used as predicates, comparable? The most important difference between adjectives and verbs is that 97,9% of the adjectives are simple, 1-place structures (i.e. they have no number code), while only 26.2% of the verbs have such a simple 1-place structure (i.e. they have an I-code). The complementation of adjectives that have a number code has the following correspondences in verb complementation:

3: Adj: anxious _A [B3]	(x ₁) (x ₂ : [Inf PRED .. (x ₁) ..](x ₂))
V: want _V [T3]	(x ₁) (x ₂ : [Inf PRED .. (x ₁) ..](x ₂))
yearn _V [I3]	(x ₁) (x ₂ : [Inf PRED .. (x ₁) ..](x ₂))
4: Adj: bent on _A [F4]	(x ₁) (x ₂ : [NOMINALIZED PRED .. (x ₁) ..](x ₂))
V: reckon on _V [T4]	(x ₁) (x ₂ : [NOMINALIZED PRED .. (x ₁) ..](x ₂))
5: Adj: insistent _A [F5]	(x ₁) (sub x ₂ : [PRED](x ₂))
V: pledge _V [T5]	(x ₁) (sub x ₂ : [PRED](x ₂))
5a: Adj: glad _A [F5a]	(x ₁) ((sub) x ₂ : [PRED](x ₂))
V: know _V [T5a]	(x ₁) ((sub) x ₂ : [PRED](x ₂))
5c: Adj: fearful _A [B5c]	(x ₁) (x ₂ : [SUBJUNCTIVE PRED](x ₂))
V: desire _V [T5c]	(x ₁) (x ₂ : [SUBJUNCTIVE PRED](x ₂))
6a: Adj: aware _A [F6a]	(x ₁) (wh x ₂ : [PRED](x ₂))
V: anticipate _V [T6a]	(x ₁) (wh x ₂ : [PRED](x ₂))
6b: Adj: sure _A [F6b]	(x ₁) (wh x ₂ : [Inf PRED .. (x ₁) ..](x ₂))
V: fix _V [T6b]	(x ₁) (wh x ₂ : [Inf PRED .. (x ₁) ..](x ₂))
7: Adj: scared _A [F7]	(x ₁) (x ₂ : [PRED _A (x ₁)](x ₂))
V: ---	

When we look at these correspondences, it soon becomes clear that, although most adjectives without a number can be compared to intransitive verbs, adjectives which do have a number code resemble transitive verbs. Two things should be noted about these correspondences: in the first place, for the only adjective which has a 7-code (*scared* [F7]), no correspondence exists. In the predicate frame that I proposed to describe the X7-code, which is the verb code that resembles F7 most, there is no co-referentiality relation between an argument of the subclause and the first argument of the matrix verb; for F7 such a co-reference relation does exist. Compare:

(172) scared_A [F7] (x₁) (x₂: [PRED_A(x₁)](x₂))
 (173) consider_V [X7] (x₁) (x₂: [PRED_A](x₂))

Secondly, the adjectives with number code 3 are compared with both I3 or T3. The comparison with I3 follows from the fact that I have proposed two possible predicate

frames for I3-verbs (Cf §3.1.3.3.). Only one of these two frames can be compared with the 3-code used for adjectives.

Although nouns are probably not used as predicates in English, 896 noun-entries have at least one occurrence of a code with a number, which indicates that the noun has complementation. In many cases, the number-code can be related to a verb code, for instance because the noun has been derived from a verb by nominalization. However, in other cases the number cannot be related to the grammatical code of some related verb, because no such verb exists. In those cases, the argument structure of nouns with a number-code resembles the argument structure of verbs only in that we also find an embedded predication.

3.4.2. Frequencies

At the beginning of this chapter, I remarked that the data on frequency shown in the tables in the sections on verb-codes, noun-codes, and adjective-codes, could not be taken as indications of how frequently these codes were used in verbal interaction. However, these numbers do tell us something about the general behaviour of verbs, nouns and adjectives. For instance, they tell us that the verb types T1 and I0 are very frequent in English. In total, there are 13427 verbs which have at least one occurrence of one specific code.¹³ Of these 13427, 6769, or 50.4%, have at least one occurrence of code T1, and 3513, or 26.2% have at least one I0-code. The total distribution of the codes is:

T1	I0	D1	L9	X9	rest	
50.4	26.2	3.7	4.2	3.5	12	%

Of all 31016 nouns that have at least one occurrence of a certain noun code, 19421, or 62.6%, are countable (code C, with or without a number), and 8339, or 26.9%, are uncountable (code U, with or without a number). Only 3256 (10.5%) nouns have at least one of the other codes. A total of 896 (2.8%) nouns have a code which indicates that the noun may have some kind of complementation (i.e. they have a code with a number). As we saw in the discussion on the noun codes, many nouns which have a complementation code are nouns which have been derived from verbs (or adjectives) by a rule of predicate formation.

There are 8634 adjectives which have one or more occurrences of a specific adjective code. Of these, 6764, or 78.3%, have a B-code (with or without a number); 1144, or 13.2% have an A-code; 627, or 7.3% have an F-code; 89 or 1% have an E-

code; and in only 10 cases, or 0.1% do we have a GU-code. In 182 cases, or 2.1%, the code indicates that the adjective is used with some kind of complementation.

3.4.3. The relation between LDOCE codes and FG structures

As we saw in the preceding sections, the LDOCE grammatical coding system leads us some way, but not all the way towards basic predicate frames and term structures. It told us something about the number of arguments and the internal structure or nature of these arguments. However, it also appeared that

- 1) Some LDOCE codes covered more than one FG structure (e.g. V3)
- 2) Sometimes, more than one LDOCE code described one FG structure (e.g. X1 and X7)
- 3) Some LDOCE codes are not used in an FG lexicon (e.g. most L-codes)

The situation that there is no one-to-one correspondence between the LDOCE codes and FG structures can partly be explained by the fact that LDOCE and FG are totally different systems of description. LDOCE's big letter codes give information about 'the way a word works in a sentence or about the position it can fill' (Preface, p. xxviii). It is, basically, a system that describes sentence patterns by giving information on the type and order of constituents in a sentence. FG predicate frames are more abstract constructs: it is not until fully specified predications are formed and expression rules have been applied that the actual linguistic expression exists. Predicate frames and term structures, for instance, contain no information about the position a word takes in a sentence. Putting the elements in a sentence in the right order is a matter that is handled in the expression rules (see fig. on page 3).

We can clearly see the difference between the LDOCE and FG systems when we look at the LDOCE V3-code, which is represented in the Table of Codes as: 'v + one object + verb form with *to*'. Although the sentence:

(174) We read the book to learn its contents

has a structure which falls under the pattern for V3 (*read* = v; *the book* = object; *to learn its contents* = verb form), the verb *read* is not coded V3 in LDOCE. The reason for this seems clear: *The book* and *to learn its contents* do not form a unit. That this is a requirement for the V3-code can be gathered from the Preface to the dictionary. The V-code is used for 'verbs that are followed, in position II (i.e. directly following the verb, H.V.), by a 2-part DIRECT OBJECT. The first part is a nounlike expression, and the second part is an INFINITIVE with or without *to*, an *-ing*-form, or a past participle' (p. xxviii). Verbs like *want* fulfil this requirement:

(175) want_v [V3] (x₁) (x₂: [Inf PRED] (x₂))

However, in some of the FG frames that I proposed to cover the V-code, the nounlike expression and the infinitive do not function as a unit at all:

- (176) see_v [V2] (x₁) (x₂) (x₃: [Inf without *to* PRED .. (x₂) ..] (x₃))
 (177) tell_v [V3] (x₁) (x₂) (x₃: [Inf PRED.. (x₂) ..] (x₃))
 (178) see_v [V4] (x₁) (x₂) (x₃: [NOMINALIZED PRED.. (x₂) ..] (x₃))

and can, therefore, never function as object together:

- (179) * The man leave was seen (by me)
 The man was seen to leave by me
 (180) * He to go was told
 He was told to go
 (181) * He coming was seen
 He was seen coming

As can be seen from these examples, the LDOCE and FG-systems are sometimes hard to compare. The only thing that justifies the rules that have been proposed in this chapter to rewrite LDOCE information as FG-structures is the fact that in many cases simple rewrite rules lead to correct FG-structures. However, until the lexicon produced by my rules has been thoroughly checked, it will not be clear whether my attempt to deduce number and type of arguments from grammatical codes has been worthwhile.

As we saw under 3) above, a number of entries in LDOCE are not included in the FG lexicon, but should be treated in grammatical rules. Examples are the copula(s), the causatives, the modals, and also all words with wordclasses that are not in the FG lexicon, such as determiners and prepositions. When I started working on this paper, I had no idea that the number of elements that had to be treated in the grammar would be so large. The verbs *be* and *become*, for example, are not the only linking verbs in LDOCE: there are 132 linking (L-code) verbs (not including the L9-verbs, which are mostly intransitive). Examples are *turn*, *form*, *end up*, etc. The number of prepositions is 165. There are 12 [I2]-verbs (modals), 81 [I3]-verbs, 9 [I4]-verbs, etc. etc.

It is clear that the greater the number of elements to be treated, the more rules are needed to keep them all apart. The grammar would need to have some very powerful mechanisms indeed to distinguish all copulas, prepositions, etc. and use each of them appropriately. This makes me wonder whether all these elements should be handled in the grammar. If this is not the case, then there are three options:

- 1) most of the elements should be in the lexicon (except elements like *be*, for which enough proof has been given that it belongs to the grammar);
- 2) there is a continuum which ranges from 'grammatical treatment' to 'lexical treatment' in which each element has a place.¹⁴ One test for the position of each item in this

continuum is the amount of meaning that the item carries (e.g. *be* has (almost) no meaning, and should thus be at the grammatical end of the continuum);

- 3) each grammatical rule, such as copula support, has at its disposal its own 'grammatical lexicon'. In the 'grammatical lexicon' for copula support, for example, all linking verbs would be found. The advantage of having such a grammatical lexicon is that the selection of a linking verb does not have to be handled by rules only. Sometimes a language user is allowed to make a choice from the grammatical lexicon to satisfy certain communicative needs. A language user might, for instance, wish to select *form* instead of *be* in the context 'they --- a nice couple'.

Notes to Chapter 3

- ¹ This description is a slightly adapted version of the description given in the Preface to LDOCE. Appendix 2 contains a copy of the relevant parts from the Preface.
- ² Whenever a co-referential argument (in this predicate frame indicated by x_2) is found in the subordinate clause of one of the predicate frames in this section, it will have to be assigned Subject function in the construction of a PREDICATION. Compare:
- (1) She hates hitting him:
 $\text{hate}_v (\text{d1}x_1: \text{she}_P (x_1))_{\text{AgSubj}} (x_2: [\text{hit}_v (x_1)_{\text{AgSubj}} (\text{d1}x_3: \text{he}_P (x_3)_{\text{GoObj}}](x_2))_{\text{GoObj}}$
- (2) He hates being hit (by her):
 $\text{hate}_v (\text{d1}x_1: \text{he}_P (x_1))_{\text{AgSubj}} (x_2: [\text{hit}_v (\text{d1}x_3: \text{she}_P (x_3)_{\text{Ag}} (x_1)_{\text{GoSubj}}]_{\text{GoObj}}$
- In (1), the argument in the subclause which is co-referential with an argument in the matrix clause is an AgentSubject, whereas in (2), it is GoalSubject.
- ³ Throughout this paper, I use the term 'ditransitive' to indicate structures involving an Agent, a Recipient, and a Goal.
- ⁴ Other functions than Location may be found for other multi-verbs.
- ⁵ They have probably been coded T2 by mistake, as can be seen from the examples which illustrate these codes:
- he came into a fortune; to come into fashion
the government came out against the new British plane
to flick the switch
- ⁶ In the QUERY-files pure intransitives can be found by searching for elements that have an I-code and no other codes, and which do not have any complement words. We find 779 verbs, among which are: *arrive*, *matter*, *elapse*, *function*, *pop up*, and *sit back*. Verbs that can be both transitive and intransitive with the same meaning cannot be found using the QUERY-files, since all reference to meaning has been removed in this file. When we search for verbs that have both an I-code and a T-code, we also find entries for which these codes are found with different senses.
- ⁷ However, *say*, *elect*, and *expect* have code T3, rather than V3 in LDOCE
- ⁸ It would be interesting to see whether an entry that has both [V3] and [T3] in the same meaning, as is the case with *want* but not with *tell*, always has a two-place V3-frame. However, I cannot check this, because my QUERY-files do not allow me to search for a combination of codes that belong to the same meaning.
- ⁹ See also Dik (1979: 138), where reportive verbs and verbs of feeling or volition are distinguished.
- ¹⁰ The verb *start*, as well as many other verbs which have aspectual meaning, has in addition to a T4 code also a T1-code, which indicates usage of the verb as Aktionsart. E.g.:

How shall we start the meeting?
We started at nine o'clock

- 11 With nouns and with adjectives a Big letter code may be used on its own: no number need follow.
- 12 Of course, verb, adjectives and nouns may also be used as input to rules of predicate formation, rather than in PREDICATION CONSTRUCTION or TERM FORMATION.
- 13 This does not mean that there are 13427 verbs in LDOCE: in fact, there are only 8649. This difference can be explained as follows. When a verb has codes T1 and IO, for instance, it will be counted twice: once because it has at least one occurrence of T1 and again because it has at least one occurrence of IO.
- 14 This continuum is synchronic. It does not imply that elements may move from one place to another on the continuum. However, it is quite possible that, over time, elements do move along the continuum, as has been shown by Goossens (1985).

Chapter 4. The Complement words

4.1. Introduction

LDOCE often gives what I call complement words with the grammatical codes. I will illustrate these with the following example: the lemma *appear* has the code [it+I6a (as if)] in one of its senses. This means that, in this sense, it has to be used in a context like:

(1) It appears (as if) she will win

The + in the code shows that *it* is obligatory, the brackets that *as if* is optional. I call *it* and *as if* complement words. Complement words may be pronouns, prepositions, determiners, adverbs, elements like *well*, *ill*, or *open*, etc.

The entry *give* in LDOCE has as its complement word the preposition *to*, which means that it can be used in the following context:

(2) He gave a book to her

Prepositions suggest the semantic function of arguments. With *give* the case is clear: the preposition *to* indicates a Recipient argument. In this chapter I want to investigate whether complement words in LDOCE can be used to predict semantic functions.

4.2. Semantic functions and prepositions in FG

The following list shows which semantic functions are used in FG, and by which preposition(s), if any, they can be indicated in English:

Function	Preposition
Agent	<i>by</i>
Positioner	<i>by</i>
Force	<i>by</i>
Processed	-
∅	-
Experiencer	<i>by, to</i>
Goal	-
Recipient	<i>to</i>
Beneficiary	<i>for</i>
Instrument	<i>with</i>
Comitative	<i>with</i>
Possessor	<i>of</i>
Location	<i>in, on, etc</i>
Direction	<i>to, into, etc</i>
Source	<i>from</i>
Time	<i>after, until, during, etc</i>
Complement (Cf Heikens 1982)	<i>on, about</i>

The prepositions *by* and *to* may, besides being used to indicate Agent/Positioner/Force/Experiencer and Experiencer/Recipient, respectively, also be used to indicate Location, Direction or Time.

In Functional Grammar, not every semantic function can be found on every argument position. The following possibilities are listed in Dik (1980: 7-9):

ACTIONS	Agent
	Agent Goal
	Agent Goal Recipient
	Agent Goal Direction
	Agent Goal Source
POSITIONS	Positioner
	Positioner Location
	Positioner Goal
	Positioner Goal Location
PROCESSES	Processed
	Force
	Force Goal
	Force Goal Recipient
	Force Goal Direction
	Force Goal Source
STATES	∅
	∅ Location

It can be gathered from this list that the first argument in a predicate frame can be an Agent, Positioner, Force, Processed or it may have ∅-function. However, it can never have the function Goal, for instance. Similarly, the second argument may be Goal or Location, but not an Agent.

We can combine the facts from these two lists. The semantic function of the argument of an intransitive verb, for instance, could only be indicated by the preposition *by* (which indicates an Agent, Positioner, or Force). If any other preposition is given, this cannot be taken to indicate the function of the argument. Following the same line of thought: if we come across a grammatical code in LDOCE which has complement word *on*, this complement word points towards a term which has the function Location.

Before I proceed, I want to make several things clear:

- a preposition like *on* indicates Location. One cannot be sure, however, whether the term that has the function Location will be an argument or a satellite.
- although the preposition *by* can be used to express Agent terms, Positioners, etc., it is unlikely that LDOCE will give *by* as a complement word to indicate these semantic functions: in the unmarked active case in English, Subject function will be assigned to the Agent (or Positioner, etc.). Subject function "masks" the expression of the semantic function, and therefore *by* will usually not be expressed.

- In the discussion so far, I have assumed that the complement words given in LDOCE indicate prepositions regularly associated with a certain semantic function. It may happen, however, that LDOCE lists a preposition that would not normally be associated with a certain semantic function, but which is, in that exceptional case, associated with this function. For instance, it might be argued that in an expression like: 'He asked for a cigarette' *for* exceptionally indicates the Goal. In the following sections I will investigate whether prepositional complement words can be used to predict semantic functions or not.

4.3. Complement words and intransitive and transitive verbs

The prepositional complement words do not always give us information on the semantic function of one of the arguments. This can be seen in particular when we look at the use of the complement word *for*. Take, for instance, *fish*, which is coded [I0 (for)]

(3) He is fishing for compliments/for trout

Fish is an intransitive verb, i.e. it has one argument. However, *for* may not be taken to indicate that the first and only argument of *fish* is a Beneficiary or has some other function expressed by *for*, since, as we saw in §4.2., the first argument of an I0-verb can only have Agent, Pos, Proc, Fo, or have \emptyset -function (it has Agent, in this case). Rather, *for* indicates the function of a satellite, probably of purpose in this case ('in order to get compliments/to catch trout'). I suggest therefore that *for* should not be included in the predicate frame of *fish* in connection with its argument.

The same situation as with *fish* occurs very often in Longman. The total number of verbs that have at least one meaning which is both intransitive and has a complement word is about 1100. In none of the cases that I have seen does the complement word function in the predicate frame.

Let us see what happens with transitive verbs. Eg.

- (4) *arm* [T1 (with)]:
They armed themselves with broken bottles
- (5) *blind* [T1 (to)]:
His desire to do it blinded him to the difficulties
- (6) *match* [T1 (for)]:
Nothing matches this hotel for good service
- (7) *divide* [T1 (by)]:
I divided the number by three

It is clear that the complement words in these examples have nothing to do with the Goals. The Goals, which are, as we would expect, not indicated by a preposition, are: *themselves*, *him*, *this hotel*, and *the number*.

However, it is not easy to decide how the constituents that start with the complement words are to be handled:

- does *with*, in the first example, indicate an Instrument?
- does *to* indicate Direction in the second example?
- what is the semantic function of *for good service* in the third example?
- as *by* does not indicate an Agent in the last example, what function does it indicate?
- is the prepositional phrase introduced by the complement word an argument or a satellite in the predications in these examples?

The total number of T1-verbs with complement words is about 1200. Which prepositions do we expect to find on the arguments of T1-verbs? The first argument may, but probably will not be indicated by the preposition *by*; the second argument can only be Goal, and Goal is not indicated by a preposition. Although the function Location can be found with certain verbs on the second argument, these verbs do not have grammatical code T1, but L9. Indeed, as we saw in the examples above, the complement words have nothing to do with the first argument of these verbs. They cannot be attached to the second argument of the verb either.

Verbs with prepositional complement words resemble prepositional verbs, and verbs with adverbial complement words resemble phrasal verbs. However, LDOCE did not list them as such. The reason for this might either be that the combination of verb plus complement word does not occur frequently enough, or that the meaning of the verb occurring by itself did not differ from the meaning of the verb when used with a complement word. Anyway, probably the best way to view the combinations of verbs with complement words is as collocations, that is, as (frequently) co-occurring elements. There seems reason for me to, at least, preserve the complement words, even though they will not be used to deduce semantic functions of arguments.

4.3.1. One argument too few

Though this seems a safe way of dealing with the complement words, it is not the case that this approach will always lead to correct predicate frames. As we saw above, T1-verbs with complement words can usually be changed into two-place predicate frames (on the basis of T1 alone) and we may or may not want to include the complement word somewhere in the frame. However, with the verb *ask* [T1 (for)] the situation is different. In:

(8) I asked him for a cigarette

a cigarette is the goal of *ask*, and *him* is the recipient of the question.¹ Rather than analyse *ask* as a two-place predicate on the basis of T1, we should analyse it as a three-

place predicate with *for* (exceptionally) attached to the goal argument. In the case of *ask* we cannot simply disregard the complement word and rewrite the T1-code as a two-place structure since this will lead to the wrong predicate frame, with one argument missing.

4.3.2. One argument too many

The opposite situation occurs with the verb *blow*, which is coded [D1 (for)] by Longman. I assume (LDOCE does not give any examples to illustrate it) that this code is used to handle contexts like:

(9) He blew a beautiful glass for me

Although this verb is coded D1, *for* in this case indicates the role of Beneficiary associated with *me*. Beneficiary, however, is not an argument, but a satellite in the predication underlying this example. We are, in fact, dealing with the two-place predicate *blow*, and not with a three-place predicate. On the basis of code [D1 (for)] *blow* gets one argument too many.

We have seen two cases involving the preposition *for* that led to problems (examples (8) and (9) above). This might suggest that it would be necessary to check all *for*-complement words, and maybe other complement words as well, to see whether they cause trouble or not. This would imply a lot of manual work: the preposition *for* occurs about 277 times as a complement word in LDOCE. As we saw in Chapter 1, detailed investigations of this kind are avoided in this paper. Fortunately, in many cases the correct predicate frame will be assigned if we only take into consideration the letter-and-number code and not the complement word:

(10) canvass [I0 (for)] He canvassed for votes (1-place)

(11) empanel [T1 (for)] They empanelled him for the jury (2-place)

4.4. Interim Conclusions

What has struck me in my investigation of the complement words so far is:

- 1) They may give information on either the semantic function of an argument or the semantic function of a satellite. In [D1 (to)] found with *give*, the complement word shows us the semantic function of an argument, but in [I0 (for)] found with *ring*, the complement word indicates the semantic function of a satellite.
- 2) Prepositions alone will usually not give enough information to predict semantic function. One and the same complement word may indicate various semantic functions. The preposition *to*, for instance, is used not only on arguments with Recipient function, but could also be used on arguments with the function Direction. Compare also:

- (12) ask for a cigarette (Goal?)
- (13) ring for the cook (Purpose?)
- (14) sing a song for me (Beneficiary)
- (15) nothing matches this hotel for good service (??)

(Part of) the reason for this ambiguity is, of course, that prepositions like *to* and *for* themselves have many different senses (LDOCE lists 25 different senses for *to* and 27 different senses for the preposition *for*).

- 3) LDOCE often does not give examples containing the preposition in question, so that it is hard to see which meaning/context is meant.
- 4) Verbs with complement words resemble phrasal/prepositional verbs.
- 5) Errors are very hard to detect: E.g. *plan* [I0 (for)] 'She hadn't planned for so many guests'. Is *plan* really intransitive here or is it a transitive verb?

4.5. Verbs with D-codes: complement words *do* indicate semantic functions

In the previous section it was shown that in many cases prepositional complement words could not be used to predict the semantic function of arguments in a predicate frame. Most examples given in that section involved intransitive and transitive verbs. However, the situation looks less hopeless with verbs that have a D-code in LDOCE. The number of D-verbs with complement words (171, against about 1100 intransitive and about 1200 transitive verbs) allows one to investigate this group in much more detail and much more fully than was possible with the intransitive and transitive verbs.

4.5.1. *To*

Of the 171 verbs with a D-code that have one or more complement words, 75 have the complement word *to*. Whereas *to* used in combination with the other codes will be left out of the predicate frame that is constructed, with D-codes *to* is, as far as I can see, always used to indicate Recipient function.

- (16) *adjust* [T1 (to)]
He adjusted himself quickly to the heat of the country
- (17) *award* [D1 (to)]
They awarded the prize to him

Thus, *adjust* will be analysed as a two-place predicate and nothing is said about the function of its arguments; *award* will be analysed as a three-place predicate with semantic function Recipient on its third argument.

The class of D-verbs with *to* in LDOCE consists of the following verbs:

accord, assign, award, bear, begrudge, bequeath, bring (back), cable, cause, (con)cede, deal, deliver, dole out, entrust, extend, forward, give (back), grant, hand out, issue, leave, lend, loan, mail, make, obtain, owe, pass, pay (back), play, pose, post, preach, present, proffer, quote, read, recommend, refund, reimburse, remit, render, rent, repay, return, score, sell, send serve, show, signal, sing, sling, slip, take, teach, telegraph, (tele)phone, telex, tell, throw, toss, vouchsafe, wave, will, wire, wish, write.

I will compare this list with the list of verbs of communication that have a 3-place structure with a Recipient given in Heikens (1982). A verb which is listed as a three place structure involving an Agent, a Goal and a Recipient in Heikens, and which has a D-code in LDOCE, will get √ while all other verbs will be marked X.

X affirmer = affirm, assert	X déclarer = declare, announce
X annoncer = announce	√ demander = ask
√ apprendre = teach	X dire = say
√ assurer = assure	√ écrire = write
√ causer = cause	√ enseigner = teach, show
X commander = command, order ²	X expliquer = state
X communiquer = communicate	X exposer = expound
X confesser = confess, acknowledge	X exprimer = express
X confier = confide	X indiquer = indicate
√ conseiller = advise ³	√ lire = read
X crier = cry, shout	√ montrer = show
X objecter = object	√ recommander = recommend
X ordonner = order (see note 2)	X répliquer = reply, rejoin
X parler = speak, talk	X répondre = answer, respond
X permettre = allow ⁴	X reprocher = reproach
X prédire = predict	X rétorquer = retort
√ promettre = promise	X révéler = reveal
X proposer = propose	X riposter = riposte
X publier = proclaim, publish	√ signaler = signal
√ raconter = tell ⁵	√ télégrapher = telegraph
X rapporter = report	√ téléphoner = telephone
X réciter = recite	

Surprisingly, only 15 out of Heikens' 43 verbs have a D-code in LDOCE, and 10 of those 15 (all except *assure*, *advise*, *order*, *promise*, and *ask*) have complement word *to*. Most verbs that do not have a D-code, like *say* and *cry*, cannot be used as follows:⁶

(18) *He said me these words

(19) *He cried me to come

However, they can be used with the preposition *to*:

(20) He said these words to me

(21) The woman cried to her child: 'Don't do that'

This use is regarded as transitive by LDOCE, for these verbs all have T-codes. In FG they are seen as ditransitives,⁷ however, so the predicate frame that these verbs get on the basis of their LDOCE code is not correct. They would have to be corrected by hand.

4.5.2. For

Examples in which a D-code with complement word *for* is used are:

- (22) Can you cash this postal order for that old lady please?
 (23) He saved his strength for an effort in the last minute of the race

As we can see in these examples, *for* indicates a Beneficiary satellite and not an argument. This means that, in the predications underlying these examples, we are dealing with transitive rather than ditransitive verbs: both *cash* and *save* have two arguments and not three. This is the case for all D-verbs. It might be concluded that [D1 (for)] must be changed into a T1-code. In all these cases, however, a T-code is already present, so this does not have to be added. I will throw away the D-code plus *for* whenever *for* is the only complement word used and I will throw away *for* alone when the D-code has other complement words as well. This means that I will throw away [D1 (for)] for the following verbs:

assure, blow, boil up, book, cash, charge, choose, crochet, cut (off), do, find, fix, get, knit, mix, net (2x), obtain, order, pack, pardon, pick, pluck, prepare, prescribe, procure, pull, purchase, revenge on, roll, root out, run off, rustle up, save, secure, snag, stand, and win

Only the complement word *for* will be left out with:

bear, blame, bring, build, buy, carve, cause, leave, make, notch, offer, owe, pay back, play, pose, pour, provide, recommend, score, sing, and take

4.5.3. With plus other complement words

The complement word *with* occurs 20 times with D-verbs together with the complement word *on*. This combination signals pairs like:

- (24) We caked the wall with mud/ We caked mud on the wall

It has been argued (Cf Dik, 1980: Ch 2) that the first of these two examples is derived from the second by a rule of predicate formation:

COMPLETIVE VERB FORMATION IN ENGLISH

input: $\text{pred}_v (x_1)_{\text{Ag}} (x_2)_{\text{Go}} (x_3)_{\text{Loc}}$
 output: $\text{pred}_v (x_1)_{\text{Ag}} (x_3)_{\text{Go}} (x_2)_{\text{Instr}}$

The 20 cases in which *with* is found together with *on* are:

cake, daub, emblazon, emboss, engrave, heap, impress, inscribe, lay, plaster, saddle, scatter, serve, shower, smear, spatter, spread, stamp, strew, and top-dress

It seems justified to say that these are all examples of verbs that allow COMPLETIVE VERB FORMATION.⁸ This means, however, that only one of the two prepositions needs to be used in creating a predicate-frame for these verbs, namely the preposition *on*. This preposition will, in these cases, result in a frame such as that shown after 'input' above. An example is, $\text{cake}_v (x_1)_{Ag} (x_2)_{Go} (x_3)_{Loc}$. A frame which provides for the constructions with the preposition *with* must not be included in the lexicon since it can be derived from the frame with a Loc argument by a rule.

On is not the only preposition which may be used to indicate the Loc function on the third argument, however. We also find the preposition *in*. This is used nine times in combination with the preposition *with*, namely with the verbs: *drape, embed, imbed, inculcate, inscribe, inset, inspire, set, and strew*. An example of a pair involving *with* and *in* which shows COMPLETIVE VERB FORMATION is:

- (25) You inspire me with admiration
 (26) He inspires hate in me

For eight of the nine verbs COMPLETIVE VERB FORMATION seems to be involved. With the verb *drape*, however, the prepositions *with* and *in* do not indicate COMPLETIVE VERB FORMATION as can be seen from the example:

- (27) Let us drape this picture of our leader with/in the national flag

The prepositions *with* and *in* can be both be used in this context with the same meaning, so no derivation is involved. However, I will not single this example out in order to correct it manually. Mentioning the exception here must suffice for now.

Yet other prepositions which are used are *over, around, and round*. Verbs which may have constructions with *with* and *over* are *drape, dredge, dust, scatter, smear* and *strew*. There is one verb which allows *with* and *around*, namely *drape*, and two which allow *with* and *round*, namely *drape* and *gird*.

There does not necessarily have to be a Loc function on the third argument of the input of COMPLETIVE VERB FORMATION, however. The argument may have the function Direction. Verbs involved with their respective prepositions are:

infuse, inject, inscribe, inset: into
dust: onto
saddle: upon⁹

We may conclude that the rule of COMPLETIVE VERB FORMATION may have both Location and Direction function on the third argument of the input. The above data might also be used to argue that the functions Location and Direction are not really two distinct functions, but are subclassifications of a more general function, which can be described by the term "Spatial". I will not go into this question here, but it may be worthwhile to devote some research to it in the future.

We also find cases where the preposition *with* is used in constructions which have parallel constructions with *to*. This is found with *entrust*, *intrust*, *issue*, *present*, but COMPLETIVE VERB FORMATION is not involved here, because no element of completion is involved.

Notes to Chapter 4

- ¹ I assume that no distinction needs to be made between the function of *a cigarette* and that of *a question* in the following two sentences:

I asked him for a cigarette
I asked him a question

- ² *Order* does not have a D1-code in the right meaning. It has a [D1 (for)] in the meanings:
'advise as necessary': He ordered three weeks rest for her
'ask for something to be made, etc': he ordered three suits for himself

- ³ *Counsel* does not have a D-code.

- ⁴ *Allow* has a D-code, but not in the sense of permit. The sense to which the D-code applies is 'to give, esp. money or time: *My father allows me some money for books*'.
Permit does not have a D-code

- ⁵ *Relate* does not have a D-code.

- ⁶ *Command* and *answer* are examples of verbs that can be used without *to* as well:

The general commanded his men to attack the city
She answered me this: '...

- ⁷ These ditransitives do not allow Object function to be assigned to the Recipient.

- ⁸ In a number of cases, though, I am not so sure whether, when *with* is used, the Goal will necessarily be **completely** covered (physically or psychologically). Examples of sentences where this doubt arises are:

They served a summons on him
They served him with a summons
He saddled me with the responsibility
He saddled the responsibility on me
They embossed the name and address of the firm on the paper
They embossed the paper with the name and address of the firm

In the first two examples, involving *serve* and *saddle*, it is hard to see what happened to the person who acts as the Goal. In the case of *emboss* ('to cause to appear by pressing') the paper may come out completely covered by names and addresses, but it does not seem unlikely that one name and address would be enough to use *with* as well.

- ⁹ Prepositions like *upon* can often be used both as a preposition indicating a Location and as a preposition indicating Direction. This is no longer a problem if we allow both Direction and Location on the third argument of the input of the rule of COMPLETIVE VERB FORMATION.

Chapter 5. Selection Restrictions

5.1. Introduction

LDOCE uses a system of semantic codes to classify each sense of an entry. These semantic codes consist of ten positions, found in each line identified by number 7, three of which may be indications of selection restrictions. The semantic codes used in these three positions can be found in Appendix 1. The three positions are used, respectively, "to classify semantic restrictions on:

- 1) a) the subject of a verb
 - b) a noun
 - c) the noun qualified by an adjective
- 2) a) the object or nominal complement of a Vi, Vl, Vt, or Vv verb
 - b) the first object or nominal complement of a Vd or Vx verb
- 3) the second object or nominal complement of a Vd or Vx verb" (LDOCE information on computer codes)

I will include the semantic codes as selection restrictions at the appropriate places in the predicate frames. It should in principle be the case that each verb with either a D-code or an X-code has semantic codes in all three positions; each verb with a T-code has codes in two positions, and each verb with an I-code or L-code, each noun and each adjective has a code in the first position only. Whenever this is not the case, either the people who compiled LDOCE have forgotten to put a code in the appropriate position (which means there will be one code too few), or we are dealing with a sense of a verb with more than one grammatical code. For instance, when we have a sense with grammatical codes T1 and D1, and semantic codes in all three positions, both T1 and D1 will get three semantic codes. This means there will be one code too many for the T1 code. The third semantic code may be disregarded for the T1-code.

The inclusion of the semantic codes given in LDOCE in the structures in an FG lexicon does not need justification. However, although this type of semantic information will certainly be interesting, it is not clear what the value of selection restrictions is in a full-blown linguistic model. In this chapter, I will address the question of how effectively selection restrictions can be used in linguistic theory. In contrast to the preceding chapters, I will spend a large part of this chapter on a theoretical discussion. This discussion may pose more questions than it answers. This does not mean that selection restrictions should not be used. In fact, as we have seen above, I will use them in the FG lexicon. It is important, however, to be aware of the advantages and the disadvantages of working with selection restrictions.

5.2. Theory

The need has been recognized long ago to distinguish between deviant and non-deviant sentences. Usually, a distinction is also made between grammatical and lexical deviance. An example of grammatical deviance is:

(1) * The boys is in the garden

and an example of lexical deviance is:

(2) * The flower walked away

In the first example a grammatical rule is violated (namely, number agreement between subject and verb); in the second example a co-occurrence restriction is violated (*walk* must have an animate subject). The latter type of deviance has often been handled by making use of selection restrictions. Selection restrictions involve reference to inherent features. When a verb is specified with a certain selection restriction on its subject, in the process of sentence formation it will have to be checked whether the restriction is compatible with the inherent features of the subject.

Although according to some linguists structures containing non-matching selection restrictions and features are ungrammatical, others (e.g. Huddleston, 1976: 14) believe that these structures are grammatical, but trigger special interpretation strategies. This latter view is shared by Dik (1978: 44 ff), and Aarts (1976).

5.3. Selection Restrictions in Functional Grammar

In FG, the structure of predicate frames and terms makes it possible to specify the selection restrictions on the argument positions to which they belong. Thus, the verb *give* has the following predicate frame:

(3) $give_v(x_1: \langle human \rangle(x_1))_{Ag}(x_2)_{Go}(x_3: \langle human \rangle(x_3))_{Rec}$

which shows that the Agent and the Recipient of *give* are human and the Goal has no restriction. When a predication is formed, and terms are inserted into the predicate frame of *give*, tests are needed to make sure that non-human terms will not be inserted in the Agent and Recipient positions. These tests involve going through the meaning of the selected terms. Suppose we select *bachelor* to go into Agent position. This entry is defined as:

(4) $bachelor(x_1) =_{df} unmarried(x_1: man(x_1))$

It is clear that we cannot conclude directly from this entry whether *bachelor* may be inserted into the Agent position, which has the selection restriction $\langle human \rangle$, or not. To

find this out we have to go through the meaning chain of *man* in the definition of *bachelor*. *Man* is defined as

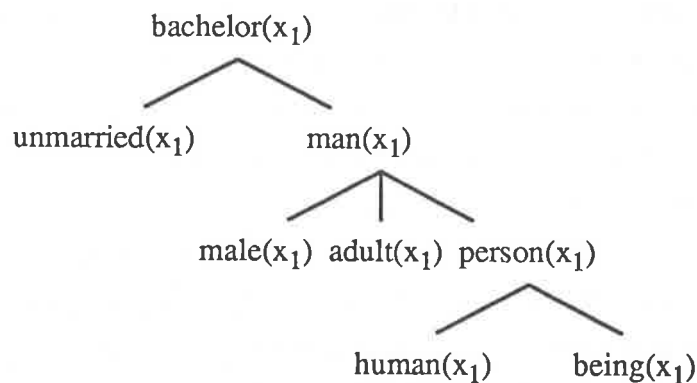
$$(5) \text{man}(x_1) =_{df} \text{male}(x_1 : \text{person}(x_1))$$

and again, we will continue to *person*

$$(6) \text{person}(x_1) =_{df} \text{human}(x_1 : \text{being}(x_1))$$

At this point we find the desired the element *human*. We can now conclude that *bachelor* may, indeed, be inserted in the Agent position of *give*.

We may represent the meaning chain of *bachelor* as follows (Cf Dik, 1978b: 35)

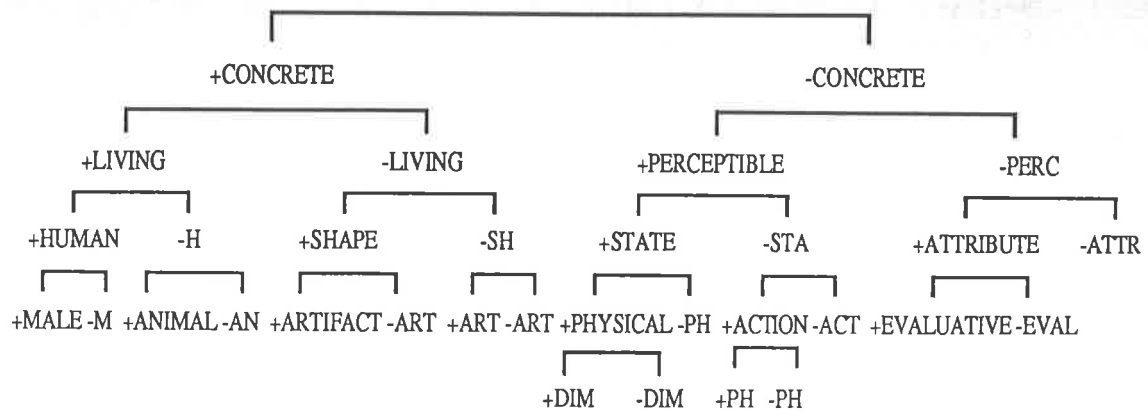


We can see that we have to go quite a way down the chain to arrive at the feature <human>. This feature is not found in the definition of *bachelor* itself, as it would be in componential analysis, but it is found in the chain created by stepwise lexical decomposition.

The entry *give* in LDOCE has 3 semantic codes: H (human) for the Agent, Z (unmarked) for the Goal and H (human) for the Recipient. This is exactly the information we need for our FG lexicon.

5.4. Aarts

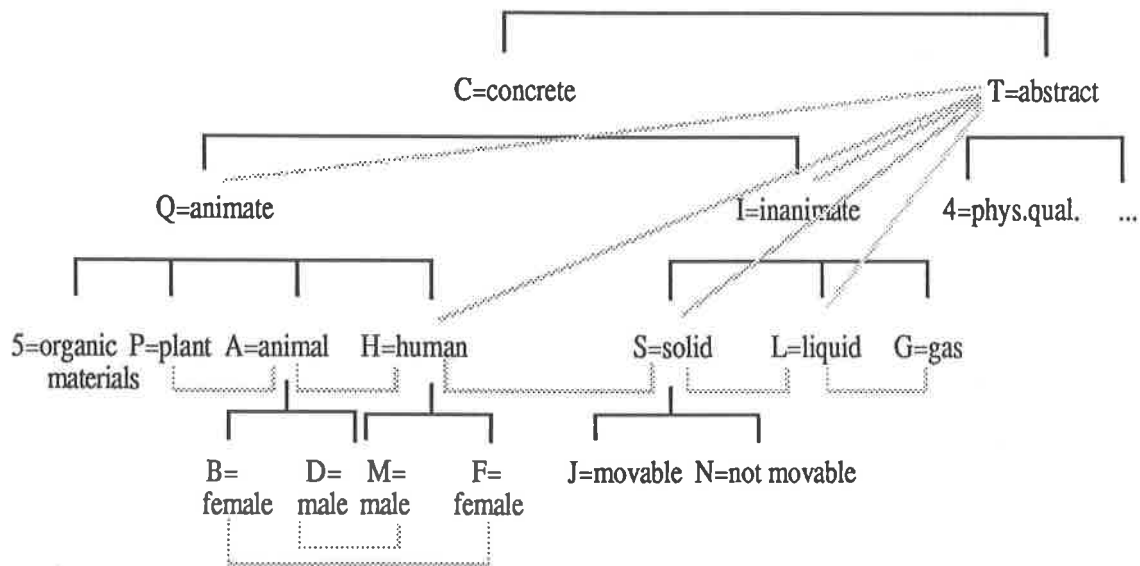
Aarts (1976) has investigated the problem of how to describe adjective-noun combinations. His proposal to treat these structures involves making use of a number of different types of features. The first of these types are the Higher Primary (in short: HPRIM) features, of which he has fourteen, and which may all be used with either a positive or a negative value (they are binary). These HPRIM features are ordered hierarchically as follows:



Beside these HPRIM features he has Lower Primary (LPRIM) features, generative features and secondary features. LPRIM features are features like [\in professional], [\in food], [\in communicate], etc. LPRIM features are always dominated by HPRIM features. We find, for instance, that examples of [\in communicate] dominated by [+ACT, +PH] are *frown*, *gesture*, etc and examples of this LPRIM feature dominated by [+ACT, -PH] are *voice*, *answer*, etc. Because HPRIM features and LPRIM features may function as contextual features, I will go into these features in the following sections to see what relation they bear to the selection restrictions in FG and the semantic features in LDOCE.

5.5. HPRIM features and Semantic codes

A number of the semantic codes used in LDOCE are used as HPRIM features by Aarts. The semantic codes used in LDOCE can be represented as follows:¹



The dotted lines in this figure are indications of combined codes:

E = S+L; K = D + M; O = A + H; R = B + F; V = P + A; W = T + I; X = T + H; Y = T + Q; 1 = H + S; 2 = T + S; 6 = T + L; 7 = G + L. The U-code, which stands for 'collective and animal or human', is not included in this figure.

When we compare the systems employed by Aarts and LDOCE, we find the following correspondences (x is 'no correspondence'):

LDOCE semantic code	HPRIM-feature
C = concrete	[+CONCRETE]
T = abstract	[-CONCRETE]
Q = animate	[+LIVING]
I = inanimate	[-LIVING]
H = human	[+HUMAN]
V = plant or animal + 5=organic mat.	[-HUMAN]
M = male and human	[+MALE]
F = female and human	[-MALE]
A = animal	[+ANIMAL]
P = plant + 5 = organic mat.	[-ANIMAL]
S = solid	? [+SHAPE]
S = solid + 7 = gas and liquid	[-SHAPE]
x	[±ARTIFACT, ±SH]
4 = physical qualities	[+PERCEPTIBLE, +STA, +PH,-DIM]
x	[-PERC]
x	[±ATTRIBUTE]
x	[±EVALUATIVE]
x	[-STATE]
x	[±ACTION]
x	[±PH]
x	[+DIM]
B = female animal	x
D = male animal	x
E = solid or liquid	x
G = gas	x
L = liquid	x
J = movable	x
N = not movable	x
O = animal or human	[+H] or [-H, +AN]
K = male animal or human	x
R = female animal or human	x
U = collective	x
W = not animate	[-CONCRETE] or [+CONC, -LIVING]
X = not concrete or animal	[-CONCRETE] or [+CONC, +LIV, +H]
Y = abstract or animate	[-CONCRETE] or [+CONC, +LIVING]
Z = unmarked	x
1 = human and solid	[+CONC, +LIV, +H] or [+CONC, -LIV,+SH]
2 = abstract and solid	[-CON] or [+CONC, -LIV, +SH]
3 = 'it' as subject or object	x
6 = liquid and abstract	x
7 = gas and liquid	x

Five things struck me in particular in the comparison of LDOCE's semantic codes and Aarts' HPRIM features:

- 1) The LDOCE code 3 does not seem to belong in the system of semantic codes at all. This code means that *it* may be used with the entry as subject or object. This is not semantic information.
- 2) A number of distinctions are made in each system which are not made in the other system (E.g. [\pm ARTIFACT] in Aarts; J and N (movable and not movable) in LDOCE).
- 3) Whereas LDOCE is slightly more specific in the subdivision of [concrete] (even though they do not have the distinction between [\pm ARTIFACT]), Aarts is vastly more specific in the [-CONCRETE] section of the above picture. In fact, LDOCE hardly subdivides the feature [Abstract] at all.
- 4) LDOCE allows a number of combinations of codes (e.g. E = solid or liquid) which are not found in Aarts.
- 5) Sometimes comparison between the semantic codes and the HPRIM features is difficult: Aarts make a distinction between [\pm SHAPE] and LDOCE distinguishes S (solid), L (liquid) and G (gas) under I (inanimate). Probably S (solid) may be related to [+SHAPE], for everything that has shape must be solid, but solid may also occur as a subcategory of [-SHAPE], as can be seen from examples like *wood* and *iron*. Aarts and LDOCE seem to subdivide the fields in different but overlapping ways.

I will conclude this section with three tables: Figure 1, Figure 2, and Figure 3. Each cell in each table gives the number of entries in LDOCE (specified on the horizontal axis) that have at least one occurrence of a semantic code (found on the vertical axis). The selection restrictions found on the vertical axis specify different properties of the entries in each table. The selection restrictions in the first table (Figure 1) specify 1) the subject of a verb, 2) a noun or 3) the noun qualified by an adjective. The selection restrictions in the second table (Figure 2) specify the object or nominal complement of Vi, VI, Vt, or Vv verbs, or the first object or nominal complement of Vd, or Vx verbs. The restrictions in the third table (Figure 3) specify the second object or nominal complement of Vd or Vx verbs.

The numbers in these figures **cannot** be interpreted as indications of the frequency of elements which have a certain semantic code in verbal interaction. What can we learn from these tables? From the first table we can learn, for instance, that 10639 noun-entries have at least one occurrence of the semantic code which tells us that the noun is abstract. This is 34,8% of the total of 30569 noun entries which have at least one occurrence of a certain semantic code. From the second table we can learn that there are no verbs with a D-code which have a first object (first object is used to indicate the Recipient of ditransitive verbs) which is restricted to elements with the feature P (Plant). An example of what we can learn from table 3 is that there are 109 verbal entries with a D-code which have code T (abstract) for the second object (used for the Goal argument of

ditransitive verbs). Inspection of the QUERY-files has made it clear that the semantic code T is often used to mark embedded predications. I will leave it to the reader to draw further conclusions from these tables. It will often be necessary to know for which entries a certain semantic code is used. These can be found using the QUERY-files which I created for this paper.

There is one point I want to stress before anyone starts investigating the semantic codes in more detail, namely that it is important to keep the picture in mind which shows the hierarchical structure of the semantic coding system used in LDOCE. This picture can be found on p 67 above. The position of a semantic code in this hierarchy may have consequences for the interpretation of one's findings. I will give two examples.

Suppose, for instance, that one would like to know how many nouns in LDOCE are human. The number of nouns which can be used for humans is **not**, as one might conclude from Figure 1 (column 1, semantic code H), 3022. In order to get the correct number, one has to add the numbers of nouns which have one of the following semantic codes:

- 1) Female and Human (F);
- 2) Male and Human (M);
- 3) Animate or Human (Q);
- 4) Abstract or Human (X);
- 5) Male (Animal or Human) (K);
- 6) Animal or Human (O);
- 7) Female (Animal or Human) (R);
- 8) Collective (Animal or Human) (U);
- 9) Human and Solid (1);

The nouns indicated by the semantic code H are only a subset of the total set of nouns which can be used to refer to humans.

As another example, let us look at all abstract nouns. From Figure 1, we might conclude that abstract nouns are far more frequent in LDOCE than human nouns. This is true, but the comparison is not really valid: the number of nouns that has semantic code T is so large because the T-code is high up in the hierarchy and is hardly subdivided. The H-code for humans is lower on the hierarchy and also subdivided for male and female.

Class:	Nouns	Adjectives	I-verbs	L9-verbs	T-verbs	V-verbs	D-verbs	X-verbs
Semantic Code								
A	1006	80	81	12	41	0	0	7
B	22	6	10	1	2	0	0	0
C	337	314	210	47	250	0	3	22
D	26	2	2	0	3	0	0	0
E	241	68	8	3	13	0	0	0
F	397	30	26	2	16	0	0	0
G	95	16	5	1	1	0	0	0
H	3022	1173	2066	232	4649	212	400	395
I	39	42	22	4	33	0	0	1
J	5091	327	168	55	95	0	0	9
K	2	3	1	0	1	0	0	0
L	608	76	37	23	21	0	0	4
M	740	28	16	0	31	0	0	1
N	1851	159	28	8	23	0	0	0
O	71	133	196	77	234	11	8	39
P	718	57	25	1	2	0	0	0
Q	26	55	36	5	34	1	0	4
R	3	2	4	0	4	0	1	0
S	1208	295	59	8	61	1	0	7
T	10639	2807	259	42	409	10	19	4
U	634	69	166	13	325	12	10	10
V	21	7	2	0	2	0	0	0
W	98	107	19	3	32	0	2	0
X	264	1817	213	27	764	37	42	42
Y	47	142	29	5	46	1	4	3
Z	417	1422	406	77	918	39	58	67
1	79	57	190	41	394	5	11	39
2	1116	618	58	12	91	1	2	7
3	1	4	24	1	13	2	0	0
4	543	83	15	1	14	0	1	1
5	1173	234	42	6	18	0	0	0
6	12	4	3	0	3	0	0	0
7	22	2	6	0	2	0	0	0

Figure 1. Semantic codes used for:

- 1) the subject of verb,
- 2) a noun, and
- 3) the noun qualified by an adjective.

The codes used with verbal entries are related to their grammatical codes. Thus, a verb might have a human subject when used transitively, and an abstract subject when used intransitively (e.g. 'the storm abated' (I0); 'He abated the nuisance' (T1)).

	T-verbs	V-verbs	D-verbs	X-verbs
A	120	0	1	3
B	6	0	0	0
C	497	1	19	62
D	0	0	0	0
E	81	0	8	10
F	23	0	0	0
G	9	0	0	0
H	1630	150	184	132
I	52	1	5	4
J	837	1	34	132
K	2	0	0	0
L	121	0	3	8
M	7	0	0	3
N	293	0	3	12
O	207	9	6	18
P	58	0	0	2
Q	31	1	1	0
R	1	0	0	0
S	514	0	15	52
T	2271	39	132	77
U	147	7	3	4
V	12	0	1	0
W	103	8	12	3
X	565	50	37	30
Y	61	3	1	2
Z	951	55	80	100
1	120	0	6	18
2	399	10	35	17
3	4	0	0	1
4	35	0	0	3
5	251	0	2	21
6	1	0	0	0
7	11	0	0	1

Figure 2 Semantic codes for:

- 1) the object or nominal complement of a Vi, Vl, Vt, or Vv verb and
- 2) the first object or nominal complement of a Vd or Vx verb.

	D-verbs	X-verbs
A	1	0
B	0	0
C	13	0
D	0	0
E	3	0
F	0	0
G	0	0
H	120	8
I	1	0
J	5	0
K	0	0
L	0	0
M	0	0
N	8	2
O	2	0
P	0	0
Q	0	0
R	0	0
S	8	2
T	109	12
U	5	0
V	0	0
W	11	0
X	29	4
Y	1	0
Z	49	18
1	2	0
2	23	2
3	0	0
4	0	1
5	2	0
6	0	0
7	0	0

Figure 3. Semantic codes used for the second object or nominal complement of a Vd or Vx verb

5.6. LPRIM features and Subject Codes

A number of semantic codes used in LDOCE are found not as HPRIM features in the system of Aarts, but as LPRIM features (Aarts, 1976: 31-34). Thus we have:

dominated by	LPRIM feature	LDOCE-code	examples from Aarts
[-AN]	[\in tree]	P = plant	cedar, birch, elm
[-SH]	[\in solid]	S = solid	wood, iron, concrete
	[\in liquid]	L = liquid	water, rain, lava

Among the LPRIM-features used by Aarts, we also find the LPRIM feature [\in food], which is used to specify all things that are edible, and the LPRIM-feature [\in writing] for words like *letter*, *book*, etc. Dik (1978b) uses these examples to show that selection restrictions sometimes have to be very specific in order to specify the correct group of words that may go into a certain argument position. The LPRIM features [\in food] and [\in writing] are specific enough for the verbs *eat* and *write*, and the total set of LPRIM-features may help us a long way toward assignment of the appropriate restrictions to all words. Unfortunately, Aarts does not give a complete list of LPRIM-features, however.

At this point, it occurred to me that I might be able to get further by using the so-called subject codes, which, like the semantic codes, are found in LDOCE on computer tape, but not in the dictionary in its printed form. These subject codes are indications of the semantic field to which a word belongs. Thus, the lemma *cell*, for instance, belongs to the field of medicine and biology and gets subject-code MD. In this way, about 120 main fields are distinguished, many of which are subdivided.

Another semantic field, or subject field as LDOCE prefers to call it, is that of 'food', marked by the subject code FO. This is where the relation between subject codes and LPRIM features comes in, for Aarts uses, as we saw above, the LPRIM feature [\in food]. The question now is: can we use subject codes, like LPRIM features, as contextual features, or, in other words, can we use subject codes as selection restrictions? I do not think this can be done without further research, and not before a number of changes have been made. There are several reasons for this.

In the first place, the subject code FO is used for a wider range of words than the LPRIM feature [\in food]. FO, when used with nouns, indicates not only things that are edible, but also other words concerning 'cooking, cheeses, dietetics, flour milling, meat packing & cutting, spices, TV dinners, food processing, canning, tea, coffee, meals, baking' (Longman documentation on the computer codes, p. 25). An example of an entry with subject code FO which does not indicate something edible is the noun *baker*. It might be objected that this entry could be automatically excluded from the set of words indicating edible things, because it has a semantic feature H (human). This is correct. However, there are also non-human entries with subject-code FO which do not indicate

edible things, for instance *dish*. The entry for *dish* has the same subject code (FO) and the same semantic code (J) as, for instance, the entry for *cake*, but *dish* is obviously not edible and *cake* is. Thus the field FO, which is subdivided for cooking, baking, and food processing, would also need to have a subdivision for 'things which can be eaten'. Other subject codes may, likewise, have to be subdivided.

Secondly, I do not know whether the list of main fields used by Longman is meant to be exhaustive or not. In fact, I do not think it is, since even the field of colour terms, which is often used as an example of a semantic field in textbooks on semantics, is not found among the main fields, nor among the subdivisions.

Thirdly, the relation between selection restrictions and semantic fields is not at all clear. One can probably say that selection restrictions and semantic fields cross-classify. It may not be concluded, however, that semantic field indicators can be used, like selection restrictions, to delimit classes of items that may occur in certain argument positions.

I conclude that the subject codes used in LDOCE are too problematic for me to include in my lexicon. They are included in the QUERY-files, so they can be used for further investigation.

5.7. Conclusion

In this chapter, we have seen that grammatical theories make a distinction between lexically deviant and lexically well-formed structures. Selection restrictions have often been used to distinguish these two types. In Functional Grammar, selection restrictions are found in the argument positions of predicate frames.

The semantic codes that Longman uses to mark semantic restrictions will be used as selection restrictions in the predicate frames which I will create. In a comparison between these semantic codes and HPRIM features used by Aarts, it was found that for some codes LDOCE provides more information, whereas for others Aarts does. Both systems provide quite a detailed system of semantic restriction.

Using (an extended version of) LDOCE's semantic coding system will not solve all problems concerning selection restrictions, however. The problem is that more restrictions are needed. To overcome this difficulty, Aarts introduced the LPRIM feature. I have shown that some of LDOCE's subject codes, like FO (food), might be used like LPRIM features to delimit groups of lexical items. If we regard a meaning definition as a hierarchical construct as shown for the word *bachelor* above, the LPRIM features and LDOCE's subject codes can be found at a much higher point in the hierarchy than the HPRIM features and LDOCE's semantic codes. If both are used as selection restrictions,

then selection restrictions could work either on a very general level ([concrete]), or on a specific level ([food]).

The subject codes cannot simply be used as selection restrictions, however, since the relation between selection restrictions and subject codes (indicating semantic fields) is not clear. Moreover, as we saw above, it is not clear whether the list of subject codes provided by Longman is complete and whether it has been properly subdivided.

All these questions remain unanswered.

The idea that most people who 'believe in' selection restrictions have, is that as soon as all (and only) the necessary selection restrictions have been found all problems will be solved. It is clear that finding all these necessary restrictions is not an easy matter. The restrictions [+HUMAN] and [+ANIMATE], for instance, can be found anywhere in the literature on selection restrictions. Aarts uses 14 binary HPRIM features and he lists more than 70 LPRIM features in describing adjective-noun combinations. However, the latter list consists only of some examples (Aarts, 1976: 31). The complete list cannot be found². LDOCE has 33 semantic codes. If the subject codes are going to be of any use as selection restrictions, we have to take into account at least 120 distinctions. The problem is that I do not know if there is a maximum to the number of selection restrictions (e.g. the number of selection restrictions may not exceed the number of lexical items of the language).

Another related question is whether there is a minimum to the number of words that may be selected from the lexicon on the basis of some selection restriction. If this is not the case, a selection restriction could be used to indicate, for instance, that in one sense the verb *blow* must have *nose* as its Goal (namely, in the idiomatic expression *blow one's nose*). In this case, however, there hardly seems to be any reason for distinguishing between the lexical item *nose* and the selection restriction [NOSE]. Where such a small number of words (here: one) may be selected, we should no longer speak of selection restrictions, but of idioms. Where, exactly, one stops speaking of idioms and starts introducing selection restrictions is not clear to me.

A number of problems related to selection restrictions might be solved, in any case, if we introduce rules like the one proposed in *Stepwise Lexical Decomposition*, which says that all Goals are subject to the general restriction:

$$\Phi (x_1)_{Ag} (x_2)_{Go}$$

x_1 can Φ
 x_2 can be Φ -ed

However, would the problem not only be moved from the area where selection restrictions are used to the area of knowledge of the world?

Notes to Chapter 5

- ¹ This is a modified version of the figure of Box Codes 9 + 10 given in the LDOCE documentation on Computer Codes
- ² Aarts (1976: 48) mentions that the number of features would be much larger if entire nominal groups were taken into account. I wonder what will happen when clauses or sentences are considered

Chapter 6. Conclusion

6.1. Constructing an FG-lexicon: Summary

In this paper, I have tried to create sets of basic predicate frames and term structures of a lexicon of Functional Grammar. The structures that I proposed are based on information from the *Longman Dictionary of Contemporary English*. The information from this dictionary (which we have on computer tape) which was relevant was:

- 1) headword
- 2) part of speech (only nouns, adjectives, verbs, phrasal verbs, prepositional verbs and phrasal prepositional verbs were used)
- 3) grammatical codes
- 4) semantic codes

This information was used in the construction of predicate frames and term structures by general rules. These rewrite rules were based on an investigation into the types of entries that have certain grammatical codes in LDOCE. The main thing we could learn from this investigation was that LDOCE provides a detailed system of classification of its entries, which was very useful in the construction of an FG-lexicon. The subcategorization of verbs, in particular, allowed us to provide verbal predicate frames with quite specific information about their argument structure, especially where embedded predications were concerned. However, it was also found that there was no 1-to-1 correspondence between FG-structures and LDOCE-codes.

The structures thus created were extended by selection restrictions, which were not extracted from the grammatical codes, but from the semantic codes given in LDOCE. The question was asked whether the so-called subject codes from LDOCE (indicating the semantic field to which an entry belongs) could be used to supplement the semantic codes. It was concluded that this could not be done without further research into the relation between semantic restrictions and subject codes.

What was lacking in the structures constructed so far were the semantic functions of the arguments. In chapter 4, an attempt was made to deduce semantic functions from the complement-words given with the grammatical codes in LDOCE. Thus, from the code [D1 (to)] the semantic functions Agent, Goal and Recipient were deduced, and from [D1 (on, with)] and [D1 (into, with)], etc. Agent, Goal and Loc/Dir were deduced.

It is not impossible that other semantic functions could be deduced from information given in LDOCE if we looked at it in more detail. Close inspection might also bring to light points where the rewrite rules proposed in Chapter 3 will fail. The dictionary contains a wealth of information that I left unexplored. However, I think that I

have extracted about as much as was possible using general rules and avoiding manual work and inspection of large numbers of entries.

6.2. *Constructing an FG-lexicon: The EXTRACT_LONGMAN_FG Program*

In chapter 2, we saw that the EXTRACT_LONGMAN_FG program was written to create a set of QUERY-files. These QUERY-files contain all information from the *Longman Dictionary of Contemporary English* which I considered relevant for this paper. They contain each entry from LDOCE which has a wordclass which figures in the FG-lexicon together with its grammatical codes, its semantic codes and its subject field codes. The QUERY-files were used to check the contents of the dictionary and to test the rules that were proposed in chapters 3, 4, and 5. After the investigation of the QUERY-files, some procedures were added to the program, incorporating all the rewrite rules suggested in this paper. The program ran over the entire dictionary again and produced a number of output files:

- 1) a file called PREDICATE_FRAMES, containing all basic predicate-frames of the FG-lexicon (See Appendix 6 for some examples);
- 2) a file called TERM_STRUCTURES, containing all basic terms (See Appendix 6 for examples);
- 3) a file called SUB_FRAMES, containing all subheadwords in LDOCE, most of which are derived predicates;
- 4) a file called LEFTOVER_FILE, containing all information which was not included in the files mentioned under 1), 2) and 3) above. This file contains all entries which have a wordclass which does not figure in the FG-lexicon: determiners, pronouns, suffixes, etc.;
- 5) a number of ERROR_FILES, containing messages of errors in the structure of the information on the LDOCE-tape, and messages of places where the program has run into difficulties.

6.3. *The Future*

In order to use the PREDICATE_FRAMES and TERM_STRUCTURES files as an FG-lexicon, the arguments which have not received a semantic function must be assigned the correct function. As we saw in chapter 4, not every semantic function can be found on every argument position. The first argument may be an Agent, Positioner, Processed, Force, or have \emptyset -function, but it may not have, for instance, Goal-function. Redundancies like these will make the task of assigning semantic functions considerably easier.

A representative lexicon containing fully-specified predicate frames and term structures is, in my view, indispensable in almost any kind of research in Functional

Grammar. The lexicon can be used in parsing and generating systems; it can be used, together with a monolingual dictionary of another language, to create a bilingual dictionary which can be used in translation and it can be used as an object of research in its own right. The real value of any model which tries to show how natural language 'works' can only be assessed when it incorporates a lexicon which reflects the store of lexical information that the natural language user has at his or her disposal.

APPENDIX CONTENTS

	page
Appendix 1 The semantic restrictions used in LDOCE.....	1
Appendix 2 The grammatical codes used in LDOCE (copy).....	2
Appendix 3 The QUERY-files.....	6
Appendix 4 Examples of the LDOCE grammatical codes.....	8
Appendix 5 Verbs which have an X1 code, but not an X7 code.....	14
Appendix 6 Examples of entries in the FG lexicon.....	15
Appendix 7 Table of Codes (copy).....	17

Appendix 1 - The semantic restrictions used in LDOCE

- A: Animal (not human) - sex unspecified
- B: Female animal
- C: Concrete (non abstract) - animate or inanimate
- D: Male animal
- F: Female & human
- G: Gas
- H: Human - sex unspecified
- I: Inanimate (not abstract) - solid, liquid, or gas unspecified
- J: Movable (solid)
- L: Liquid
- M: Male and human
- N: Not movable (solid)
- P: Plant
- Q: Animate (Plant, animal, or human)
- S: Solid - movable or not movable unspecified
- T: Abstract
- X: Not concrete or animal (abstract or human)
- Z: Unmarked (no semantic restrictions)
- E: Solid or liquid
- K: Male (animal or human)
- O: Animal OR Human - sex unspecified
- R: Female (animal or human)
- U: Collective & animal OR & human
- V: Plant of Animal (not human)
- W: Not animate (abstract or inanimate)
- Y: Abstract or animate
- 1: Human and solid
- 2: Abstract and solid
- 3: "it" as subject or object
- 4: physical qualities which are not associated with actual matter (i.e. solid material).
(E.g. diseases, measurements of physical dimensions, heat, etc.)
- 5: organic materials, i.e. substances such as bone, etc.
- 6: Liquid and abstract
- 7: Gas and liquid

Appendix 2 - The grammatical codes used in LDOCE (copy)

xix

17.1.2.2 Positions III and IV show the position of OBJECTS and COMPLEMENTS after a verb. Position III is the position directly after a verb, and position IV is the position that a second OBJECT or COMPLEMENT has after a first OBJECT or COMPLEMENT: thus, in the sentence *The grateful president elect gave Margaret a new car*, *grateful* is in position I, *elect* is in position II, *Margaret* is in position III, and *car* is in position IV. *New* in *a new car* is in position I. In *My trouser legs got wet*, *trouser* is in position I, *wet* is in position III, and there is nothing in positions II and IV.

17.1.2.3

POSITION

She gave him a His	I new new	coat. coat was black.
The wood was 3 inches The president	II thick. elect	visited the factory.
She was He gave the He kicked the	III happy. boy ball.	a boat.
He made him He gave the boy a	IV king. boat.	

17.1.3

[A]	=	adjectives and nouns that come in position I, before a noun. Examples: <i>the main</i> (adj) <i>difficulty</i> <i>General</i> (n) <i>Smith-Foreseeue</i> <i>a trouser</i> (n) <i>leg</i> <i>atomic</i> (adj) <i>scientists</i>
[B]	=	adjectives that can come in position I or in positions III and IV as COMPLEMENTS. (Most English adjectives are of this type, so an adjective that can fill all these positions in all its meanings is not marked at all.) Examples: <i>a happy man</i> <i>The man became happy.</i> <i>She made him happy</i>
[C]	=	nouns that can be counted, usually as UNITS (<i>dog, box</i>), but in some cases as types (<i>wine</i>). They can be used with <i>one</i> or with <i>alan</i> . (Most English nouns are of this type, so a noun that behaves in this way in all its meanings is not marked at all.) Examples: <i>alone dog</i> <i>3 dogs</i> <i>4 boxes</i> <i>the 5 best wines of France</i> <i>felt strong desires</i> <i>become a general</i>
[D]	=	verbs that are followed by 2 nouns (<i>buy, book; wall, paint</i>), PRONOUNS (<i>him, them</i>), or nounlike expressions (<i>what to do</i>), which come in positions III and IV, often as INDIRECT OBJECT (<i>the boy</i>) + DIRECT OBJECT (<i>a book</i>). These nouns, PRONOUNS, etc., always represent (REFER TO) something else, not each other. Examples: (Give the boy a book.) } [D1 (to)] (Give a book to the boy.) } (Buy him a book.) } [D1 (for)] (Buy a book for him.) } Tell the boy the truth. [D1 (to)] Tell the boy what to do. [D6b] (Spray the wall with paint.) } [D1 + with/on] It cost me \$6. [D1] (Spray paint on the wall.) } (He made her a good husband. [D1 (for)] They put the difficulty down to (v adv prep) his failure [D1])
[E]	=	adjectives, adverbs, and nouns that can come in position II. Examples: <i>the president elect</i> (adj) <i>3 years old</i> (adj) <i>3 years ago</i> (adv)

1.1.0 Besides the familiar expressions "noun", "adjective", "verb", etc., this dictionary gives more detailed descriptions of the behaviour of words in English. This information, which is in the form of letters and numbers, is shown between SQUARE BRACKETS, []. These letters and numbers are explained thoroughly in the following pages, which you should read with care if you want to get the most out of your dictionary. When you have done this, you will be able to remind yourself quickly of what the letters and numbers mean by looking at the tables inside the back cover.

17.1 THE BIG LETTERS

17.1.1 The big letters give information about the way a word works in a sentence or about the position it can fill: for example, they tell you whether a noun can or cannot have a plural, whether an adjective can come before or after a noun, or both, and whether a verb can have an OBJECT.

17.1.2 Positions

17.1.2.1 Positions I and II in the following explanations of the big letters show the position of nouns, adjectives, and adverbs round a main noun. Position I is *before* the noun: thus, in the sentence *He gave her a new car*, the adjective *new* is in position I, before the noun *car*. Position II is *after* the main noun: thus, in the sentence *She was 47 years old*, the adjective *old* is in position II, after the noun *years*.

[F]	=	adjectives and adverbs that come in positions III and IV as COMPLEMENTS (usu. after <i>be</i> or <i>seem</i>). Examples: <i>She was asleep</i> (adj). <i>He found her asleep</i> (adj). <i>She was abroad</i> (adv). <i>He found her abroad</i> (adv). <i>The meeting was yesterday/upstairs</i> (adv). <i>She was happy to do that</i> (adj). [F3] <i>I'm not sure where to go</i> (adj) [F6b]
[GC]	=	nouns that usually represent groups. In the plural they take a plural verb, but in the singular they can take either a singular or (esp. in British English) a plural verb and plural PRONOUN. Examples: <i>3 different committees</i> <i>The committee is/are angry because it is/they are divided</i>
[GU]	=	nouns and adjectives that have no separate plural form. The nouns (<i>Admiralty, left</i>) usually represent groups, and can take either a singular or (esp. in British English) a plural verb. The adjectives (<i>the accused</i>) take a singular verb and PRONOUN for one person or thing, and a plural verb and PRONOUN for more than one. Examples: <i>The Admiralty (n) is/are divided</i> . <i>The left (n) is/are divided</i> . <i>The accused (adj) is angry because he wants a new trial</i> . <i>All the accused (adj) are angry because they want new trials</i>
[H]	=	adverbs that can be used with PREPOSITIONS (<i>within, into</i>) and with other adverbs, esp. with those having the same form as PREPOSITIONS (<i>through</i>). Examples: <i>The bullet went right through (the wall)</i> . <i>It was ready well within the time</i> . <i>It fell splash into the water</i>
[I]	=	verbs, most of which are INTRANSITIVES, that need not be followed by anything in position III or IV, and need have no ADVERBIAL. An [I] verb followed by nothing at all is shown with the zero mark [Ø]. Examples: <i>We paused</i> . [IØ] <i>We smoked</i> . [IØ] <i>The sun rose</i> . [IØ] <i>The bridge blew up</i> (v adv). [IØ] <i>Yes, I can</i> . [IØ] <i>They came</i> . [IØ] <i>They came to hate her</i> . [I3] <i>They came running</i> . [I4] <i>If anyone asks you, don't let on</i> (v adv). [IØ] <i>I helped</i> . [IØ] <i>This factory is closing soon</i> [I4]
[L]	=	LINKING verbs, which are followed by something that represents (REFERS TO) the subject. This "something" may be a nounlike expression (<i>President</i> [L1]), or an adjective (<i>Jamous</i> [L7]) serving as COMPLEMENT in position III, or an ADVERBIAL (<i>here, to Harlow</i> [L9]). Examples: <i>She became President</i> . [L1] <i>She became jamous</i> . [L7] <i>She acted as (v prep) President</i> . [L1] <i>It cost £6</i> . [L1] <i>He ended up (v adv) rich</i> . [L7] <i>The car's going up the hill</i> [L9]
[N]	=	nouns that are VOCATIVES; they can be used in the singular in direct address to one person. Examples: <i>Goodbye, doctor!</i> <i>Goodbye, General!</i>
[P]	=	nouns and adjectives that are used only with plural verbs and PRONOUNS. Examples: <i>The police (n) are here</i> . [P] <i>The trousers (n) are here</i> . <i>They fit me</i> . [P] <i>The dramatics (n) of the performance were interesting</i> . [P] <i>Do the dead (adj) ever return?</i> [<i>the + P</i>]
[R]	=	nouns that are names (<i>God, the Earth</i>) or name-like (the <i>sack</i>). They are used either always with <i>the</i> or never with <i>the</i> . Examples: <i>God the Earth got the sack</i>
[S]	=	nouns that are special singular nouns and can be used with <i>a</i> or <i>an</i> but not with <i>one</i> . They cannot be counted, and have no plural form. Examples: <i>Have a think about it</i> . <i>I heard a babble of noises</i>
[T]	=	verbs, many of which are TRANSITIVES, that are followed by a noun or nounlike expression in position III as a DIRECT OBJECT which does not represent (REFER TO) the subject, unless REFLEXIVE. Examples: <i>She kicked the boy/him</i> . [T1] <i>She said "I'm here."</i> [T1] <i>She said (that) she was there</i> . [T5a] <i>He looked at (v prep) the man</i> . [T1] <i>She blew up (v adv) the bridge</i> . [T1] <i>Tell the boy</i> . [T1] <i>Tell the truth</i> . [T1] <i>They considered him</i> . [T1] <i>I helped him</i> . [T1] <i>We put off (v adv) holding the meeting</i> . [T4] <i>If anyone asks you, don't let on (v adv) that you know</i> . [T5] <i>I can't put up with (v adv prep) all that noise</i> . [T1] <i>The firm has decided to close its London branch</i> [T3]
[U]	=	uncountable nouns that cannot be counted as nouns of type [C] can. Nouns of type [U] take a singular verb and are not usually used with either <i>a/an</i> or <i>one</i> . Examples: <i>Sugar/Love is sweet</i> . [U] <i>He came by car</i> . [U] <i>Felt strong desire [U] a lot of earth [U] lots of wine [U] Dramatics is interesting [U]</i>
[V]	=	verbs that are followed, in position III, by a 2-part DIRECT OBJECT. The first part is a nounlike expression, and the second part is an INFINITIVE with or without <i>to</i> , an <i>-ing</i> form, or a past participle. Note that <i>I saw the man leave</i> [V2] (without <i>to</i>) can become, in the PASSIVE, <i>The man was seen to leave</i> (with <i>to</i>). Examples: <i>I saw the man leave</i> (= The man was seen to leave). [V2] <i>I helped him clean the windows</i> . [V2] <i>I helped the boy to do it</i> . [V3] <i>I asked the man to do it</i> . [V3] <i>I saw the man leaving</i> . [V4] <i>to have a house built</i> [V8] <i>We looked at (v prep) the man jumping</i> . [V4a] <i>They inhibited her from (v prep) doing it</i> [V4b]
[Wa]	=	Some information about the forms of words is shown by the letters [Wa] (adjectives and adverbs), [Wn] (nouns), [Wp] (PRONOUNS), and [Wv] (verbs). (Note Numbers after the letter [W] do not have the same meaning as those numbers which may appear after any of the other letters, and which are explained below.)
[Wa1]	=	Some adjectives and adverbs have irregular forms of the COMPARATIVE and SUPERLATIVE (<i>good/well, better, best</i>). These irregular forms are given with the adjectives and adverbs themselves. Most adjectives and adverbs form their COMPARATIVE and SUPERLATIVE with <i>more</i> and <i>most</i> ; (intelligent(y), <i>more</i> intelligent(y), <i>most</i> intelligent(y)). The letters [Wa] are used for adjectives and adverbs that do not belong to either of these 2 types, or that have special rules for the pronunciation of their COMPARATIVE or SUPERLATIVE.
[Wa2]	=	usually short adjectives and adverbs that form their COMPARATIVE and SUPERLATIVE in one of the following 3 ways: nice, nicer, nicest (+ -r/-st) fast, faster, fastest (+ -er/-est) angry, angrier, angriest (y becomes i + -er/-est)
[Wa3]	=	longer adjectives and adverbs that can form their COMPARATIVE and SUPERLATIVE either like those in [Wa1] or with <i>more</i> and <i>most</i> . Examples: <i>secure</i> } <i>more secure</i> <i>most secure</i> <i>large-hearted</i> } <i>larger-hearted</i> <i>largest-hearted</i> <i>more large-hearted</i> <i>most large-hearted</i> <i>well-known</i> } <i>better-known</i> <i>best-known</i> <i>more well-known</i> <i>most well-known</i>

[W4]	= adverbs in which the letters <i>-ally</i> are pronounced /li/. Examples: <i>historically</i> /hi'stɔrɪkəlɪ/ (from <i>historical</i> /hi'stɔrɪkəl/), <i>poetically</i> /pəʊ'etɪkəlɪ/ (from <i>poetic</i> /pəʊ'etɪk/).	[Wv6]	= verbs that are not usually found in tenses that use the <i>-ing</i> form, with forms of the verb <i>be</i> . Example: <i>He knows French</i> . (The code [Wv6] means that you cannot say * <i>He's knowing French</i> . Compare <i>He's learning French</i> .)
[W5]	= adjectives and adverbs that do not usually form a COMPARATIVE or SUPERLATIVE. Examples: <i>an atomic bomb</i> <i>the main problem</i>	[X]	= verbs that are followed in position III by a nounlike expression as DIRECT OBJECT and by a second expression that represents (REFERS TO) the DIRECT OBJECT. This second expression is a COMPLEMENT in position IV and may be a nounlike expression (<i>fool</i> [X1]) or an adjective (<i>foolish</i> [X7]), or an ADVERBIAL (<i>in the box, here</i> [X9]). Examples: <i>They considered him a fool</i> . [X1] <i>They regarded him as (v prep) a fool</i> . [X1] <i>They considered him foolish</i> . [X7] <i>They regarded him as (v prep) foolish</i> . [X7] <i>Put it in the box</i> . [X9] <i>Put it here</i> . [X9] <i>He made her a good wife</i> . [X1]
[Wn]	Nouns of type [Wn] are usually the names of certain animals and plants. Some of these have 2 plurals, an ordinary plural with <i>-s</i> and a plural exactly like the singular; others have only one plural, exactly the same as the singular; the numbers show these facts about the word forms (INFLECTIONS) of these nouns.		Note: An adjective without a letter is considered to be of type [B]. A noun without a letter is considered to be of type [C]. All verbs have at least one letter. Adverbs ending in <i>-ly</i> have no letters except sometimes [Wa], and few other adverbs have them.
[Wn1]	= nouns that usually change (add <i>-s</i>) in the plural, but sometimes (as with animals when talking about hunting) have a plural that is the same as the singular. Examples: <i>He photographed several lions</i> . } [Wn1] <i>He shot several lions</i> . } <i>There were 3 pheasants for sale</i> . } <i>He went looking for pheasant because he was hungry</i> . } [Wn1]		Adjectives may be of one or more of the following types: [A], [B], [E], [F], [G], [GU], [P], [Wa]
[Wn2]	= nouns that usually or often do not change (add <i>-s</i>) in the plural, but can do so (as when talking about different kinds of animal, esp. fish, with the same name, or about insects or other small animals which cause disease or damage). Examples: <i>He caught 5 salmon</i> . } [Wn2] <i>The Atlantic and Pacific salmon are closely related</i> . } <i>There were 2 quails for sale</i> . } [Wn2] <i>He shot quail to make money</i> . } <i>The plant was covered in greenfly</i> . } [Wn2] <i>There are 2 greenflies on my hand</i> . } <i>This animal is infested with hookworm</i> . } [Wn2] <i>2 large hookworms were found in his stomach</i> . }		Adverbs may be of one or more of the following types: [E], [F], [H], [W], [Wa]
[Wn3]	= nouns that never change in the plural. It is the same as the singular. Examples: <i>There was a sheep in the garden</i> . } [Wn3] <i>The garden was full of sheep</i> . } <i>He bought a grouse/3 grouse for dinner</i> . [Wn3]		Nouns may be of one or more of the following types: [A], [C], [E], [GC], [GU], [N], [P], [R], [S], [U], [Wn]
[Wp]	PRONOUNS marked [Wp] are shown in tables: All the forms of these (personal) PRONOUNS appear on page xxxvii. All the forms of these other (relative) PRONOUNS appear on page xxxviii. These letters show certain facts about the forms and pronunciation of verbs, or direct the reader to tables. All the forms of this verb (be) appear on page xxxviii. All the forms of these verbs (AUXILIARY verbs) appear on page xxxviii. = verbs in which /ə/ is not pronounced when <i>-ing</i> or <i>-er</i> is added. Examples: <i>couple</i> /'kʌpl/, <i>coupling</i> /'kʌplɪŋ/, <i>coupler</i> /'kʌplə/ (Compare <i>couples</i> /'kʌpləz/, <i>coupled</i> /'kʌpləd/.) = verbs that are often used in the <i>-ing</i> form as adjectives. Example: <i>her flying feet</i> = verbs that are often used in the <i>-ed</i> form as adjectives. Examples: <i>I can fly</i> . [T1] <i>I helped clean the windows</i> . [T1] <i>I leave the mirror</i>		Verbs may be of one or more of the following types: [D], [I], [L], [T], [V], [Wv], [X] (Note: These verb patterns are also listed in, G. Leech and J. Svartvik, <i>A Communicative Grammar of English</i> (Longman, 1975), 835-872, and you can find explanations of them in greater detail in, R. Quirk, S. Greenbaum, G. Leech and J. Svartvik, <i>A Grammar of Contemporary English</i> , (Longman, 1972) 12.29-70.) Pronouns may have the letters [Wp], directing the reader to a table.
[Wp1]		17.2 THE NUMBERS	17.2.1 The numbers give information about the way the rest of a phrase or clause is made up in relation to the word being described. They have the same meaning wherever they appear, except after the letters [Wa], [Wn], [Wp], and [Wv], which have their own rules.
[Wp2]		[0]	means that nothing need follow the verb in positions III or IV, or as an ADVERBIAL. It is used in [I0]. Examples: <i>We paused</i> . [I0] <i>We smoked</i> . [I0] <i>The bridge blew up</i> (v adv). [I0] <i>Yes, I can</i> . [I0] <i>No, I haven't</i> . [I0] <i>They came</i> . [I0] <i>I helped</i> [I0]
[Wv]		[1]	means that a verb is followed by a nounlike expression in position III [L]; [T1] or 2 nounlike expressions in positions III and IV [D1]; [X1]. Examples: <i>She became President</i> . [L1] <i>She became what she wanted to be</i> . [L1] <i>She kicked him</i> . [T1] <i>Give the boy a book</i> . [D1] <i>They considered him a fool</i> . [X1] <i>She blew up (v adv) the bridge</i> . [T1] <i>They regarded him as (v prep) a fool</i> . [X1] <i>I helped her</i> . [T1] <i>I told him</i> . [T1] <i>He put off (v adv) the meeting</i> . [T1]
[Wv1]		[2]	means that a verb is followed by the INFINITIVE without <i>to</i> (<i>fly, clean, leave</i>). Examples: <i>I can fly</i> . [T1] <i>I helped clean the windows</i> . [T1] <i>I leave the mirror</i>
[Wv2]			
[Wv3]			
[Wv4]			
[Wv5]			

means that a word is followed by the INFINITIVE with *to* (to go, to see, to climb, to believe, to please, to answer, to know).
 Examples: *I want to go.* [T3] | *I came to see the truth.* [T3] | **Tell the boy to climb the mountain** [C3] | *a year to be alone* [S3] | *an attempt to believe* [U3] | *John is eager to please.* [F3] | *John is easy to please; an easy person to please.* [B3] | *They called on (v prep) him to answer.* [V3] | *He turned out (v adv) to know the answer* [L3]

means that a verb is followed by the -ing form (singing, running, cooking, dancing).
 Examples: *I enjoyed singing/their singing.* [T4] | *She came running.* [T4] | *He watched mother cooking the dinner.* [V4] | *He put off (v adv) holding the meeting.* [T4] | *She ended up (v adv) dancing on the table* [L4]

means that a word is followed by *that* + a CLAUSE.
 Examples: *I know (that) he'll come.* [T5a] | **Tell the boy (that) it's time.** [D5a] | **Tell him so.** [D5b] | *He warned her (that) he would come.* [D5a] | *a desire (that) she (should) go* [C5c] | *Is there proof that he is here?* [U5] | *He is eager (that) she (should) go.* [F5c] | *happy (that) she did it* [F5a] | *if anyone asks you, don't let on (v adv) that you know* [T5]

means that a word is followed in position III by a *wh*-word, *how*, or *as if*, + either a CLAUSE (*who should go: why he came*) or a phrase (*where to go*). (Note that any verb marked [1] can be followed by *what* + a CLAUSE. A verb is marked [6] only if it will take the other *wh*-patterns.)
 Examples: *He decided (who) should go.* [T6a] | **Tell me (where) to go.** [D6b] | *the reason (why) he came* [C6a] | *I'm not sure (where) to go.* [F6b] | *She considered (when) to go.* [T6b] | *If anyone asks you, don't let on (v adv) where you live* [T6a]

means that a verb is followed by an adjective (*famous, foolish*).
 Examples: *She became famous.* [L7] | *They considered him foolish.* [X7] | *They regarded him as (v prep) foolish* [X7]

means that a verb is followed by a past participle (*built, trapped*).
 Examples: *to have a house built* [V8] | *He got trapped* [L8]

means that a verb needs an ADVERBIAL, and that a noun or adjective needs a descriptive word or phrase.
 Examples: *She lives here.* [L9] | **Put it in the box.** [X9] | *a film/tennis buff* [C9] | *jurisdiction over us all* [U9] | *located in Florida* [F9] | *She ended up (v adv) in China* [L9]

17.2.2 Adjectives may have one or more of the following numbers (apart from those following the letters [Wa]): [3], [5], [6], [9]
 Nouns may have one or more of the following numbers (apart from those following the letters [Wn]): [3], [5], [6], [9]
 Verbs always have one or more numbers, which may be any of those from [0] to [9]. (They may also have a number following the letters [Wv]).
 Adverbs and PRONOUNS never have a number (except for those following the letters [Wa] and [Wp]).

17.3 THE LITTLE LETTERS

[a] In [1a] and [4a], [a] means that the adverb in a *v adv* marked [T1a, 4a] or the PREPOSITION in a *v prep* marked [V4a], comes just after the verb (even before a noun).
 Examples: *They batted down (v adv) the hatches.* [T1a] | *He left off (v adv) working.* [T4a] | *He looked at (v prep) him jumping* [V4a]

[5a], [a] means that the word *that* can be left out between a word and a following CLAUSE.
 Examples: *I know (that) he'll come.* [T5a] | **Tell the boy (that) it's true.** [D5a] | *He was sure (that) she knew* [F5a]

In [6a], [a] means that a word is followed by a *wh*-word + a CLAUSE (*who should go, why he came*).
 Examples: *He decided (who) should go.* [T6a] | *the reason (why) he came* [C6a] | *I'm not sure (where) I should go* [F6a]

[b] In [1b] and [4b], [b] means that the adverb in a *v adv* marked [T1b] or the PREPOSITION in a *v prep* marked [V4b] is separated from the verb, as by the noun phrase *foreign workers* or the PRONOUN *her* in these examples.
 Examples: *They are keeping foreign workers under (v adv).* [T1b] | *They inhibited her from (v prep) doing it* [V4b]

[c] In [5b], [b] means that a verb can be used with *so*, or sometimes with *not*.
 Examples: *I believe (so/not).* [T5b] | *So I see.* [T5b] | *I told you so* [D5b]

In [6b], [b] means that a word is followed by a *wh*-word + phrase including the INFINITIVE with *to* (*whom to see, where to go*).
 Examples: *He decided (whom) to see.* [T6b] | **Tell me (where) to go.** [D6b] | *I'm not sure (where) to go* [F6b]

In [5c], [c] means that a wish, desire, order, suggestion, etc., can be expressed by a verb used with *should* or (esp. in *AmE*) used in the INFINITIVE form as a SUBJUNCTIVE. [5c] is a formal pattern.
 Examples: *I desire (that) she (should) go/not go.* [T5c] | *a desire (that) she (should) go/not go* [C5c] | *He is eager (that) she (should) go/not go* [F5c]

17.4 MEANINGS HAVING MORE THAN ONE PATTERN

17.4.0 When a word or a meaning of a word can be used in more than one pattern, all the patterns are shown together. Less important or rarer ones are enclosed within the signs ().
 17.4.1 smoke? 1 is marked *v* [10;T1]. This means both the following patterns are possible: *We smoked.* [10] | *We smoked cigarettes* [T1]
 The examples *felt a desire* [C] | *felt strong desire* [U] are shown together as *desire? n* [C;U].
 Other examples: *through a lack of water* [C] | *for lack of water* [U] = *lack? n* [C;U]
She became President. [L1] | *She became famous* [L7] = *become v 1* [L;L7]
They regarded him as (v prep) a fool. [X1] | *They regarded him as (v prep) foolish* [X7] = *regard as v prep* [X1,7]
He became a general. [C] | *General Smith-Fortescue came to tea.* [A] | *Have some more tea, General* [N] = *general? n* [C;A;N]
Dramatics is an interesting art. [U] | *The dramatics of her performance were interesting = dramatics n* [U;(P)]
If anyone asks you, don't let on (v adv). [10] | *Don't let on (v adv) that you know.* [T5] | *Don't let on (v adv) where you live.* [T6a] | *Don't let on (v adv) how to open it* [T6b] = *let on v adv* [10;T5,6a,b]

Forgive us our sins. [D1] | Forgive us. [T1] | Forgive our sins [T1] = forgive v [D1;T1]
 17.4.2 Note: some, but not all, nouns that are titles of people and are marked [A;N] are used without *a* or *an* after [L] and [X] verbs: *She became chairman* [C;N;(A)]; but not **She became doctor* [C;A;(N)]

17.5 WORDS USED WITH PATTERNS

17.5.0 When a main word *must* or *may* be used with a less important word, and the less important word is *you, it, this, whether, if, to be*, an adverb (like *up*), or a PREPOSITION (like *over*), then the less important word is shown together with the

Appendix 3 - The QUERY-files

The information in the query-files is contained in QUERY-lines. Each line in the query-file corresponds with an entry in the dictionary. The lines contain several types of information, and each type has a unique reference number (the QUERY-code):

- lemma	10
- lemma which has been derived by tilde-derivation	11
- lemma which has been derived by dash-derivation	12
- also-word	1000
- also-word which has been derived by tilde-derivation	1001
- also-word which has been derived by dash-derivation	1002
- phonetic information	30
- wordclass	50
- grammatical codes with complement-words and the appropriate semantic codes	61
- codes for subject field	70

The dictionary can be inspected by using the program QUERY. I will here describe very briefly how QUERY can be used to search the dictionary.

An example of a line in the QUERY-files is:

```
abdicate 10 /"aebdikeit/ 30 V 50 IO/-from|H T 61 T1|H T 61 PL 70.
```

This is the entry for the lemma *abdicate*, which has the wordclass verb (V 50), and is pronounced /"aebdikeit/; it has grammatical codes IO (intransitive) and T1; the IO-code has complement-word *from*; both the IO and the T1 codes have semantic codes H (human) for the subject and T (abstract) for object. The subject field that this word belongs to is PL (politics).

To search for a specific type of information one has to create a pattern which consists of one or more **pairs**. A pair is a WORD followed by a QUERY-code. Thus, we can search for all entries in the dictionary that have wordclass N (noun) with the pattern

```
N 50
```

In this pattern the wordclass N is the WORD and 50 is the QUERY-code.

We can also search for **parts** of words or codes by using square brackets: []. In addition to the square brackets we can use the #, which indicates the beginning/end of the word or code. Thus, [# means that what follows is the first part of the word/code, and #] means that what precedes is the last part of the word/code. We may also use 'don't cares': * for a word and % for a code. When you use a 'don't care' anything will match.

In this way, we can search for all lemmata (whether they are also words, derived, or normal does not interest us) with the pattern

* [#1]

This pattern means: match any word (don't care *) that has a QUERY-code which starts with a 1.

More complex patterns can be created using sequences of pairs, or using the boolean operators .AND., .AND. .NOT. and .OR. For further information on the creation of patterns and the use of the QUERY-program, I refer you to the documentation about QUERY.

Appendix 4 - Examples of the LDOCE grammatical codes

Verb Codes:

[#D1] 6

bid, bung, cost, fling, give, hedge around with, insinuate into, leave, nick, plunder, promise, reach, render into, run through, show round, subject to, trust to.

[#D5] 6

advise, assure, bet, cable, caution, convince, forewarn, grant, have over, have got over, inform, instruct, notify, persuade, petition, phone, promise, remind, represent, satisfy, show, signal, stake, teach, telegraph, telex, tell, tip off, wager, warn, warrant, wire, write.

[#D6] 6

advise, ask, attribute to, dice for, dictate to, find, inform, instruct, relate to, show, teach, tell, train.

[#I0] 61

beg off, bumble, climb, countermarch, digest, emote, foot fault, go together, inhale, liberalise, moisten, parse, prevaricate, recycle, salivate, sit up, squawk, tangle, urinate.

[#I2] 6

can, could, dare, do, may, might, must, need, shall, should, would.

[#I3] 6

assent, bid fair, burst, compete, conspire, endeavour, go on, hesitate, ought, pine, scheme, send, set out, stand, unite, write back.

[#I4] 6

be, burst out, bust out, come, forbear, go, go on, write, write back.

[#I5] 6

appear, chance, come about, grizzle, happen, seem, transpire, wail, warn, write back.

[#I6] 6

appear, seem.

[#I8] 6

be, have.

[#L1] 6

aggregate, approximate to, bode, come to, cost, end up, form, give out, leave, play at, read, scale, stand, total, weigh.

[#L3] 6

be, incline, turn out.

[#L4] 6

be, come across as, end up, wind up.

[#L5] 6

be, go.

[#L6] 6

be.

[#L7] 6

become, burn, come in, come over, dye, fall, freeze, go, lie, part, pose as, sham, smell, stand, strike, taste, wax.

[#L8] 6

get.

[#L9] 6

breeze, cling, disgorge, finish, gas, handle, land, mess, pat, plod, pump, rocket, scruple, sit, speak, stooge, take on, trample, wear.

[#T1] 61

barrage, budget for, climb, cross_examine, discern, emulsify, flail, graze, incline, lay up, mismanage, oxidize, potter away, reactivate, ride on, settle for, sport, talk out, underscore, yank.

[#T2] 6

come into, come out against, dare, flick.

[#T3] 6

arrange, care, clamor, covenant, demand, disdain, essay, get, itch, make, neglect, plot, profess, refuse, scorn, stoop, try, want.

[#T4] 6

anticipate, break off, commence, continue, defy, do with, fall to, get, go without, hit on, kick against, love, pass up, preclude, reckon on, resist, see about, stick, try.

[#T5] 6

anticipate, avow, cable, complain, daresay, direct, envision, foreordain, holler, judge, move, order, premise, purpose, recommend, resolve, scream, submit, think, wish.

[#T6] 6

ask, calculate, confide, deliberate, discern, divine, establish, figure out, foretell, hash out, infer, latch onto, marvel, picture, prescribe, read, reflect, see, specify, touch on, vise.

[#V2] 6

bid, have, hear, help, know, let, look at, make, notice, observe, overhear, perceive, pray, see, watch.

[#V3] 6

assign, behove, bring in, come down on, count on, deputise, egg on, equip, groom, inspire, love, need, pester, predicate, program, recognise, school, slate, telephone, trouble.

[#V4] 6

bargain for, catch, countenance, culminate in, employ in, frighten into, hail as, hear about, introduce to, keep from, mind, plume upon, prevent, provoke into, regard as, represent as, smell, tax with, watch.

[#V8] 6

get, have, need, see, want.

[#X1] 6

adjudicate, assume, brand, colour, consider, declare, entitle, hail, know, mistake, observe, perceive, promote, put down as, represent, style, term.

[#X7] 6

assume, brush, consider, emulsion, grow, keep, make out, pass, presume, pronounce, rake, represent, sandpaper, send, slit, take, wedge.

[#X9] 6

bowl, catch, construe, drop, flash, hack, jab, lie, muffle, perch, plunk, prise, rave, scale, ship, sling, spray, suppose, thrust, volley, whistle.

Noun Codes:

N 50 - [#A#,#A/,#A] 61

aunt, cadet, chief justice, corporal, doctor, figure, giant, inspector, lowbrow, maternity, mob, northwest, parson, pocket_handkerchief, pulp, reverend, sen~ora, signor, spot, carryout, vanilla, yesterday.

[#C#,#C/,#C] 61

band, brussels sprout, church, commitment, discotheque, earwig, fable, flounder, garbage truck, guest, indisposition, kettledrum, ligature, mannerism, misanthrope, neckband, orchid, pitcher, precipice, fridge, seashell, singleton, stump, teahouse, unity, water cart, zombie.

[#C3] 61

association, challenge, commission, concession, cry, effort, entreaty, impulsions, indisposition, instinct, moment, motivation, opportunity, proclivity, proposal, resolution, signal, turn, wish.

[#C5] 61

assertion, claim, confirmation, decree, desire, fact, foreboding, hunch, intuition, myth, premise, presumption, promise, proposal, request, rider, rumour, submission, undertaking.

[#C6] 61

doubt, notion, reason.

[#C9] 61

cake, confessor, crowd, due, fauna, follower, ground, intimate, mark, mixer, notation, plot, proponent, race, regulo, rinse, service, spender, time, wallah.

N 50 - [#E] 61

avoirdupois, bey, celsius, couchant, effendi, esq., esquire, hell, junior, number one, pasha, proof, q.c., queen's counsel, regina, rex, sahib, senior, sterling, superior, troy weight.

[#GC,#GC/,#GC] 61

audience, cabal, chapel, commission, constituency, corte\ge, division, fatigue party, games, ground staff, infield, leadership, mess, neighborhood, panel, sixth form, steel band, syndicate, town, tribunal, vestry.

N 50 - [#GU#,#GU/,#GU/] 61

bar, civil service, cp, field, gentle sex, house, infantry, kremlin, liberal party, marine corps, meritocracy, nobility, polity, privy council, red crescent, rising generation, shits, timpani, upper crust, womanhood.

N 50 - [#GU9] 61

establishment, posterity, public, whereabouts.

[#N#,#N/,#N/] 61

blackguard, cabby, christ, congresswoman, crone, doctor, fellow, girl, guvnor, honor, killjoy, mac, milady, mother superior, officer, pedant, pouf, pup, rotter, sheik, slob, superintendent, czar, waiter.

[#P#,#P/,#P/] 61

basics, clergy, corn laws, flies, great, hoi polloi, jumps, little folk, menses, mumps, pampas, potteries, provinces, scripture, snuffles, suburbs, vapours, young.

[#P3] 61

qualifications.

[#P9] 61

antics, belongings, betters, conditions, dimensions, directions, environs, expenses, features, figures, folk, fruits, interests, kith and kin, lands, mansions, marching orders, moments, movements, parts, walking papers, waters, writings, young.

[#S#,#S/,#S/] 61

better, cellarage, concentration, disservice, eyeful, footing, golden handshake, and jump, kinship, lull, morbidity, pace, peep, postbag, purr, resiliency, say, shambles, soupc,on, table, upbringing, wording.

[#S3] 61

curiosity, disinclination, disposition, incapacity, necessity, need, readiness, right, rush, scurry, urge, yen.

[#S5] 61

belief, consciousness, feeling, pleasure, possibility, provision, proviso, risk, sense, theory.

[#S9] 61

cost of living, dynamic, enlargement, fear, front, hand, lot, play, press, proposition, rain, realization, rush, semblance, spirit, time, undoing, zest.

[#U#,#U/,#U/] 61

asphyxia, bounty, check, conserve, demand, effusion, fire drill, gossamer, ice water, kwela, marram grass, national insurance, partnership, popularity, radiology, roup, shellshock, stint, timbre, vino.

[#U3] 61

carte blanche, desire, disinclination, effrontery, endeavor, external evidence, grace, impulse, incapacity, inclination, intent, motivation, need, power, readiness, refusal, room, will.

[#U5] 61

anticipation, anxiety, assertion, certitude, conception, confirmation, consciousness, conviction, desire, dissatisfaction, doubt, entreaty, evidence, external evidence, faith, fear, hope, indication, possibility, probability, proof, proposal, realization, reservation, risk, speculation, supposition, testimony.

[#U9] 61

authenticity, clemency, convenience, descent, dotage, enormity, extraction, fodder, life_blood, lib, negotiation, parentage, ply, provenance, purview, relaxation, rot, society, thought, vapor.

[#R] 61

arctic circle, balance of power, big dipper, border, burden of proof, chase, city hall, holy communion, cover, dartmoor, dialling tone, draft, empty set, exodus, fleet street, functionalism, golden age, grippe, hejira, horizontal, january, king james version, lee, lower chamber, marrow, michaelmas, missis, mum, new year, numbers, old woman, moral re_armament, pathetic fallacy, big dipper, primus inter pares, puss, rat race, reserve, roadway, sagittarius, second person, solar plexus, sterling area, sunrise, ticket, trinity, upper hand, veterans day, warp, white feather, world bank.

[#R9] 61

best, bug, cost of living, Creator, earth, eve, Expo, front, like, Mach, marine, middle name, missus, missis, operation, Place, plate, Point, Proctor, Reich, Republic, riding, Road, Row, small, talk, trade, vote, world.

Pronoun Codes:

P 50 - [#Wp1,#Wp2] 61

he, her, hers, herself, him, himself, his, I, it, itself, me, mine, myself, ours, ourselves, she, that, thee, theirs, them, themselves, they, thine, thou, thyself, us, we, which, who, whom, whose, ye, you, yours, yourselves.

Adjective Codes:

A 50 - [#A#,#A/,#A/] 61

bare, candid, connubial, dire, equestrian, first_degree, full_scale, hand_to_mouth, integral, literary, mere, niggling, orthopaedic, philistine, present, pyrotechnical, riverside, serene, storybook, towering, venerable, working.

[#B#,#B/,#B/] 61

argumentative, bosomy, cliquey, counterfeit, difficult, empirical, finished, glottal, hooked, individual, jocular, lustrous, mottled, old_womanish, pestiferous, primitive, relaxing, scary, slaty, strident, third_rate, undone, victorian.

[#B3] 61

anxious, concerned, contented, difficult, easy, fit, hard, horror_stricken, horror_struck, right, set, slow, sufficient, thankful.

[#B9] 61

aware, colonial, connected, considered, grown, meaning, ordered, shod, thought_out, travelled, traveled.

A 50 - [#E] 61

broad, concerned, down, everlasting, fast, going, incorporate, light, minor, offshore, overall, plus, proximo, short, strong, ultimo, whatsoever.

[#F#,#F/,#F] 61

agog, apart, beat, bound, chewed up, conformable, desirous, drunk, fifty_fifty, friendly, hard by, indifferent, loose, numb, partial, poorly off, prompt, reliant, senior, sorry, sure, uncertain, wedged.

[#F3] 61

apt, bound, content, determined, due, engaged, frightened, glad, hell_bent, keen, like, loth, powerless, pushed, ready, right, sorry, unlikely, wont.

[#F5] 61

adamant, afraid, arguable, ashamed, aware, certain, clear, confident, conscious, convinced, desirous, eager, elated, fated, fed up, frightened, glad, happy, hopeful, indicative, insistent, mad, overjoyed, positive, proud, relieved, rumoured, rumored, scared, sorry, sure, willing.

[#F6] 61,

aware, careful, clear, doubtful, fussy, sure, uncertain, undecided.

[#F9] 61

badly_off, better, bound, clad, embosomed, embowered, enamoured, enamored, environed, favoured, favored, ill, located, minded, poised, reputed, set, situated, stuck, tenable.

A 50 - [#GU] 61

above, above_mentioned, accused, aforesaid, aforementioned, bereaved, former, insured, latter, undersigned.

Appendix 5 - Verbs which have an X1-code, but not an X7 code:

QUERY-pattern: [X1] 6 .EN. .NIET. [X7] 6

Number of occurrences: 41

acclaim, anoint, appoint, baptize, baptise, brand, brand, choose, christen, commission, confound with, constitute, count, create, crown, denominate, dub, elect, entitle, fancy, hail, hail as, know, miscall, mistake for, name, nickname, nominate, number, postmark, promote, propose, put at, put down for, seed, stamp, style, subscribe, suspect, take for.

Appendix 6 - Examples of entries in the FG-lexicon

Predicate frames:

load down-F [T1/ with] (x1: <1>(x1)) (x2: <1>(x2))
 loaded-A [B] (x1: <T>(x1))
 loaded-A [F] (x1: <H>(x1))
 loadstar-N [C] (x1: <T>(x1))
 loadstone-N [C] (x1: <T>(x1))
 load with-J [D1] (x1: <H>(x1)) (x2: <1>(x2)) (x3: <2>(x3))
 loaf-N [C] (x1: <J, T>(x1))
 loaf-N [U] (x1: <J>(x1))
 loaf-V [I0/ about] (x1: <H>(x1))
 loaf-sugar-N [U] (x1: <J>(x1))
 loam-N [C] (x1: <S>(x1))
 loam-N [U] (x1: <S>(x1))
 loan-N [C] (x1: <T>(x1))
 loan-V [D1/ to] (x1: <H>(x1)) Ag (x2) Rec (x3) Go
 loan collection-N [C] (x1: <J>(x1))
 loanword-N [C] (x1: <T>(x1))
 loath-A [F3] (x1: <H>(x1))(x2: [Inf PRED](x2))
 loth-A [F3] (x1: <H>(x1))(x2: [Inf PRED](x2))
 loathe-V [T1] (x1: <H>(x1)) (x2)
 loathe-V [T4] (x1: <H>(x1)) (x2: [NOMINALIZED PRED .. (x1) ..](x2))
 loathing-N [U] (x1: <T>(x1))
 loathing-N [C] (x1: <T>(x1))
 loathsome-A [B] (x1: <T>(x1))
 lob-V [T1] (x1: <H>(x1)) (x2: <J>(x2))
 lob-V [L9] (x1: <J>(x1)) (x2)
 lob-N [C] (x1: <T>(x1))
 lobby-N [C] (x1: <N>(x1))
 lobby-N [GC] (x1: <U>(x1))
 lobby-V [T1] (x1: <H>(x1)) (x2: <H>(x2))
 lobby-V [L9] (x1: <H>(x1)) (x2: <H>(x2))
 lobe-N [C] (x1: <5>(x1))
 lobed-A [B] (x1: <5>(x1))
 lobotomy-N [C] (x1: <T>(x1))
 lobotomy-N [U] (x1: <T>(x1))
 lobster-N [C] (x1: <A>(x1))
 lobster-N [U] (x1: <5>(x1))
 lobsterpot-N [C] (x1: <J>(x1))
 local-A [B] (x1: <T>(x1))
 local-N [C] (x1: <H, T>(x1))
 local colour-N [U] (x1: <T>(x1))
 locale-N [C] (x1: <T>(x1))
 localism-N [U] (x1: <T>(x1))
 localism-N [C] (x1: <T>(x1))
 locality-N [C] (x1: <T>(x1))
 localize-V [T1] (x1) (x2: <T>(x2))
 localise-V [T1] (x1) (x2: <T>(x2))
 local option-N [U] (x1: <T>(x1))
 local time-N [U] (x1: <T>(x1))
 locate-V [T1] (x1: <H>(x1)) (x2)
 locate-V [X9] (x1: <H>(x1)) (x2: <T>(x2)) (x3)
 locate-V [L9] (x1: <H>(x1)) (x2)
 located-A [F9] (x1)

Term structures:

(i/dnx1: Kaiser-N (x1))
 (i/dnx1: kill-N (x1))
 (i/dnx1: King James Version-N (x1))
 (i/dnx1: Kings-N (x1))
 (i/dnx1: King's English-N (x1))
 (i/dnx1: king's evil-N (x1))
 (i/dnx1: Kirk-N (x1))
 (i/dnx1: kismet-N (x1))
 (i/dnx1: knighthood-N (x1))
 (i/dnx1: Koran-N (x1))
 (i/dnx1: Kremlin-N (x1))
 (i/dnx1: kuomintang-N (x1))
 (i/dnx1: kyrie eleison-N (x1))
 (i/dnx1: kyrie-N (x1))
 (i/dnx1: Labour-N (x1))
 (i/dnx1: labour market-N (x1))
 (i/dnx1: Lamaism-N (x1))
 (i/dnx1: land-N (x1))
 (i/dnx1: larboard-N (x1))
 (i/dnx1: lash-N (x1))
 (i/dnx1: last judgment-N (x1))
 (i/dnx1: last straw-N (x1))
 (i/dnx1: last word-N (x1))
 (i/dnx1: lee-N (x1))
 (i/dnx1: left wing-N (x1))
 (i/dnx1: length-N (x1))
 (i/dnx1: Lent-N (x1))
 (i/dnx1: Lent term-N (x1))
 (i/dnx1: Leo-N (x1))
 (i/dnx1: Liberalism-N (x1))
 (i/dnx1: Libra-N (x1))
 (i/dnx1: life-N (x1))
 (i/dnx1: like-N (x1))
 (i/dnx1: like-N (x1))
 (i/dnx1: limelight-N (x1))
 (i/dnx1: limit-N (x1))
 (i/dnx1: line-N (x1))
 (i/dnx1: little woman-N (x1))
 (i/dnx1: liturgy-N (x1))
 (i/dnx1: long jump-N (x1))
 (i/dnx1: Lord-N (x1))
 (i/dnx1: Lords-N (x1))
 (i/dnx1: lowdown-N (x1))
 (i/dnx1: lower class-N (x1))
 (i/dnx1: Lower House-N (x1))
 (i/dnx1: Lower Chamber-N (x1))
 (i/dnx1: lump-N (x1))
 (i/dnx1: lure-N (x1))
 (i/dnx1: Mach-N (x1))
 (i/dnx1: macrocosm-N (x1))
 (i/dnx1: Madonna-N (x1))
 (i/dnx1: Magnificat-N (x1))
 (i/dnx1: main-N (x1))
 (i/dnx1: main chance-N (x1))

Appendix 7 - Table of Codes (copy)

WORKING PAPERS IN FUNCTIONAL GRAMMAR

Editor: Lachlan Mackenzie

Executive editors: Caroline Kroon & Rodie Risselada

WPFG publishes papers which are (a) not (yet) ready for official publication, but sufficiently interesting as contributions to on-going discussions within FG and (b) papers that will be officially published, but whose publication will take at least one year after publication in WPFG. For all information contact the following address

Caroline Kroon/Rodie Risselada
Klassiek Seminarium
University of Amsterdam
Oude Turfmarkt 129
NL-1012 GC Amsterdam
tel. - (0)20 - 525 2571

Papers can be ordered as follows :

(a) Transfer the required amount of money to postal account (giro) no. 5923629, in the name of C.H.M. Kroon -inz. WPFG-, Amsterdam;

(b) Send a (Euro-)cheque or, from abroad, an international money order written out in Dutch guilders to the editorial address.

Please specify the required items.

For standing orders contact the above address.

The following papers are now available:

1. Martin Harris - Word order in contemporary French: a functional view. Hfl 5.00.
2. Simon Dik - Copula auxiliarization: how and why? Hfl 6.00.
3. Gerard Steen - Grammar and metaphor. The consequences of an anomaly. Hfl 6.00.
4. Peter Kahrel - Some aspects of derived intransitivity. Hfl 6.00.
5. Luke Zimmermann - Subordinate clauses in Australian aboriginal languages. Hfl 7.00.
6. Liesbeth Afman - Les constructions pronominales en français. Hfl 6.00.
7. Louis Goossens - The auxiliarization of the English modals. Hfl 6.00.
8. Machtelt Bolkestein - Parameters in the expression of embedded predications in Latin. Hfl 6.00.
9. Simon Dik - Linguistically motivated knowledge representation. Hfl 6.00.
10. A. Moutaouakil - Towards an adequate representation of illocutionary force in FG. Hfl 5.00.
11. Simon Dik - On the notion 'Functional Explanation'. Hfl 6.00.
12. Casper de Groot and Machiel Limburg - Pronominal elements: diachrony, typology and formalization in FG. Hfl 6.00.
13. Albert Rijksbaron - The pragmatics and semantics of conditional and temporal clauses; some evidence from Dutch and Classical Greek. Hfl 6.00
14. Jan Rijkhoff - Word order universals revisited: the principle of Head Proximity. Hfl 6.00.
15. Lachlan Mackenzie - Aspects of nominalization in English and Dutch. Hfl 6.00.
16. Co Vet - A pragmatic approach to tense in FG. Hfl 5.00.
17. Lourens de Vries - The Wambon relator system. Hfl 6.00
18. Simon Dik - Two papers on the computational application of FG. Hfl 6.00.
19. Bieke van der Korst - Twelve sentences: a translation procedure in terms of FG. Hfl 6.00.
20. Casper de Groot - Predicate formation in FG. Hfl 6.00.

21. Jan Nuyts - Negatives are not fond of travelling. A cognitive-pragmatic reconsideration of negative raising. Hfl 6.00.
22. Kees Hengeveld - The Spanish mood system. Hfl 6.00.
23. Kees Hengeveld - A functional analysis of copula constructions in Mandarin. Hfl 5.00.
24. Hetty Voogt - Constructing an FG lexicon on the basis of LDOCE. Hfl 5.00.
25. Lachlan Mackenzie - The representation of nominal predicates in the fund: a new proposal. Hfl 5.00.
26. Evelien Keizer - Definiteness and indefiniteness: a scalar representation. Hfl 5.00.
27. Kees Hengeveld - Layers and operators. Hfl 5.00.
28. Johan van der Auwera - On the position of Dutch complementizers. Hfl 5.00.
29. Jan Rijkhoff - A typology of operators. Hfl 5.00.
30. Peter Harder - The instructional semantics of conditionals. Hfl. 5.00.
31. Jan Nuyts - Functional Procedural Grammar: an overview. Hfl. 5.00.
32. Hella Olbertz - Periphrastic Aspect in Spanish. Hfl. 5.00.

Also available are Simon Dik's 1978 monograph Stepwise lexical decomposition (Hfl 5.00) and a complete bibliography of FG books, articles and papers (Hfl 2.50). These items can be ordered in the same way as a Working Paper.

